

ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ VB SCRIPT

Περιεχόμενα

1. Εισαγωγή	3
1.1. Αναφορά	3
1.2. Ευκολία Χρήσης & Εκμάθησης.....	3
1.3. ActiveX Scripting	3
1.4. Ιστορία της γλώσσας.....	3
1.5. Πρόσθετες γλώσσες προγραμματισμού	3
2. Εισαγωγή στον Προγραμματισμό.....	4
2.1. Βασικές Αρχές του Προγραμματισμού	4
2.2. Ανάθεση	4
2.3. Σύγκριση	5
2.4. Μεταφορά.....	5
2.5. Αρχές Συγγραφής Προγράμματος.....	6
2.6. Ρουτίνες και συναρτήσεις	7
2.7. Τύποι δεδομένων	8
2.8. Δομημένος προγραμματισμός	9
2.9. Objects και Class.....	10
2.10. Έλεγχος λαθών	12
3. Αντικείμενα της VBScript	12
3.1. Dictionary object	12
3.2. Drive object.....	14
3.3. Drives Collection.....	14
3.4. File object	15
3.5. Files Collection	16
3.6. FileSystemObject Object.....	16
3.7. Folder Object	17
3.8. Folders Collection.....	18
3.9. TextStream Object	18
3.10. Παρατηρήσεις	18
4. Object Model της Altec VB Language	19
4.1. IUniScripObject.....	19
4.2. IUniscriptDesignObject	19
4.3. IUniscriptFormObject.....	20
4.4. IUniScriptClientDMObject	23
4.5. IUniScriptServerDMObject.....	24
4.6. IUniscriptTableObject	25
4.7. IUniScriptFieldObject.....	27
4.8. IUniscriptDeclarObject	28
4.9. IUniScriptSelectorObject.....	29
4.10. IUniScriptDBObject	29
5. Παραδείγματα Χρήσης των UniScript Ojects.....	31
5.1. Σενάριο 1 ^ο :	31
5.2. Σενάριο 2 ^ο :	33
5.3. Σενάριο 3 ^ο :	35
5.4. Σενάριο 4 ^ο :	36
5.5. Σενάριο 5 ^ο :	38
5.6. Σενάριο 6 ^ο :	43
5.7. Σενάριο 7 ^ο (CREATE BUTTON)	50
5.8. Σενάριο 8 ^ο (CREATE FORM).....	50

6. Debugger προγραμμάτων εφαρμογής	51
6.1. Γενικά	51
6.2. Ενεργοποίηση Debugger	51
6.3. Λειτουργία Debugger	51



1. Εισαγωγή

1.1. Αναφορά

Η Basic Scripting Edition είναι το νεότερο μέλος της οικογένειας γλωσσών προγραμματισμού Visual Basic. Χρησιμοποιείται σε ένα μεγάλο φάσμα εφαρμογών, με το να δίνει ιδιότητες προγραμματισμού σε client και Server περιβάλλοντα.

1.2. Ευκολία Χρήσης & Εκμάθησης

Η γνώση της Visual Basic είναι αρκετή για κάποιον ώστε να μπορεί να χρησιμοποιεί αμέσως τη VBScript, αλλά και για κάποιον που πρώτη φορά έρχεται σε επαφή με τον προγραμματισμό μέσω VBScript η εκμάθηση είναι εύκολη και βοηθά ταυτόχρονα στο να μπορεί κάποιος να προγραμματίζει σε όλα τα περιβάλλοντα Visual Basic.

1.3. ActiveX Scripting

Η VBScript επικοινωνεί με την κύρια εφαρμογή μέσω του ActiveX® Scripting. Με το ActiveX® Scripting η κύρια εφαρμογή δεν χρειάζεται ειδική υλοποίηση του κώδικα για κάθε script. Το ActiveX® Scripting δίνει την δυνατότητα στους κατασκευαστές προγραμμάτων να υλοποιούν δικά τους run time (Unisoft Scripting Language). Η λειτουργία του ActiveX® Scripting είναι να ενεργοποιεί ένα host στην κύρια εφαρμογή που εκτελεί τις εντολές της VBScript.

1.4. Ιστορία της γλώσσας

Η VBScript είναι ένα προϊόν της Microsoft, το οποίο δίνεται δωρεάν στους κατασκευαστές λογισμικού (προγραμμάτων) για να εισάγουν τη δυνατότητα δημιουργίας script γλωσσών μέσα στις εφαρμογές τους.

Το 1990 η Microsoft δημιούργησε την τρίτη έκδοση των Windows (Windows 3.0) και εισήγαγε για πρώτη φορά ένα χρόνο μετά την Visual Basic.

Η Visual Basic ήταν ένα εύχρηστο περιβάλλον event driven προγραμματισμού για λειτουργικά συστήματα όπως τα Windows 3.0. Η εξέλιξη των εφαρμογών γραφείου (Microsoft Office) οδήγησε στις επόμενες εκδόσεις την Microsoft να ενσωματώσει τις ιδιότητες την Visual Basic μέσα σ' αυτά.

Γεννήθηκε λοιπόν η Visual Basic for Applications η οποία ήταν μια έκδοση της Visual Basic με λιγότερες ιδιότητες όσον αφορά την ανάπτυξη εφαρμογών αλλά αξιοποιούσαν όλες τις δυνατότητες των εφαρμογών γραφείου.

Στην εξέλιξη της τεχνολογίας και με την έλευση του Internet και των multi tier προγραμμάτων δημιουργήθηκε η ανάγκη ενσωμάτωσης ευφυΐας σε client προγράμματα όπως ο Internet Explorer όπου στο client tier θα εκτελούνταν λειτουργίες χωρίς την ανάγκη επικοινωνίας με τον Server.

Έτσι η Microsoft δημιούργησε την VBScript η οποία είναι πλέον τμήμα πολλών λειτουργικών συστημάτων όπως τα Windows 2000, NT, ME καθώς και του Mac OS 8.0.

Σήμερα η VBScript βρίσκεται στην έκδοση 5.8 και εξελίσσεται συνεχώς κυρίως χάριν της ανάγκης πιο ολοκληρωμένων υπηρεσιών στον Internet Explorer.

1.5. Πρόσθετες γλώσσες προγραμματισμού

Επιπλέον της VB Script η εφαρμογή Atlantis υποστηρίζει τη χρήση πρόσθετων γλωσσών προγραμματισμού όπως:

- Java Script
- Perl
- Rexx

- Tcl
- Python
- Lua

Η γλώσσα προγραμματισμού μπορεί να διαφοροποιείται (από τις διαθέσιμες) για κάθε πρόγραμμα που εκτελείται στην εφαρμογή. Η δήλωση για τη χρησιμοποιούμενη γλώσσα προγραμματισμού γίνεται στην πρώτη γραμμή κάθε προγράμματος με το παρακάτω τρόπο:

```
@Language="Γλώσσα"
```

π.χ. @Language="VBScript"

Όπου το "Γλώσσα" μπορεί να λάβει μία από τις παρακάτω τιμές:

VBScript	(για τη γλώσσα VB Script)
Jscript	(για τη γλώσσα Java Script)
PerlScript	(για τη γλώσσα Perl)
ObjectREXXScript	(για τη γλώσσα REXX)
TclScript	(για τη γλώσσα Tcl)
Python	(για τη γλώσσα Python)
LuaScript	(για τη γλώσσα Lua)

Η δήλωση επιλογής γλώσσας μπορεί να προστεθεί στο πρόγραμμα και υπό τη μορφή σχολίου -πάντα στη πρώτη γραμμή του προγράμματος- όπως συντακτικά ορίζει η δηλωθείσα γλώσσα π.χ. για Java Script:

```
// @Language = 'JScript'
```

Αν δεν υπάρχει δήλωση γλώσσας στη πρώτη γραμμή του προγράμματος τότε εννοείται ότι χρησιμοποιούμενη γλώσσα είναι η VB Script.

2. Εισαγωγή στον Προγραμματισμό

2.1. Βασικές Αρχές του Προγραμματισμού

Η ιστορία της εξέλιξης των υπολογιστών είναι συνδεδεμένη με άπειρες ανακαλύψεις οι οποίες όμως στην ουσία στηρίζονται σε τρεις πολύ απλές αρχές. Η ανάθεση, η σύγκριση και η μεταφορά είναι όλη η φιλοσοφία στην οποία στηρίζεται όλος ο κόσμος της πληροφορικής (συμπεριλαμβάνονται και το Hardware και το Software).

2.2. Ανάθεση

Είναι η δυνατότητα να δίνεται μια τιμή σε μία μεταβλητή. Τα παραδείγματα που ακολουθούν θα χρησιμοποιούν τις αρχές προγραμματισμού σε VBScript.

```
X = 5
```

Δίνουμε μια αρχική τιμή σε μία μεταβλητή

```
X = 3
```

```
Y = X * 6
```

Το X παίρνει την τιμή 3 και το Y παίρνει την τιμή $X * 6 = 18$.

```
X = 3
```

```
X = X + 2
```

Εδώ υπάρχει το κύριο παράδοξο του προγραμματισμού όπου φαινομενικά εξομοιώνεται το X με το $X + 2$. Η λειτουργία της ανάθεσης βασίζεται στην αρχή Η ΤΙΜΗ ΠΟΥ ΒΡΙΣΚΕΤΑΙ ΣΤΗΝ ΔΕΞΙΑ ΠΛΕΥΡΑ ΤΟΥ ΙΣΟΝ, ΑΦΟΥ ΓΙΝΟΥΝ ΟΙ ΠΡΑΞΕΙΣ

ΠΟΥ ΠΕΡΙΛΑΜΒΑΝΟΝΤΑΙ ΑΝΑΤΙΘΕΤΑΙ ΣΤΗΝ ΑΡΙΣΤΕΡΗ ΠΛΕΥΡΑ. Έτσι στο παράδειγμά μας το X θα πάρει την τιμή $3 + 2 = 5$.

$X = \sin(30)$

Το X παίρνει την τιμή της συνάρτησης ημίτονο, $\sin(30) = 0,5$.

$X = 2$

$Y = 3$

$Z = (X * 7) + (Y * 4)$

Ο προγραμματισμός ακολουθεί (όπου είναι εφικτό) τις αρχές των μαθηματικών κι έτσι εδώ θα γίνουν πρώτα οι υπολογισμοί μέσα στις παρενθέσεις και μετά το άθροισμα των παρενθέσεων $Z = 14 + 12 = 26$.

2.3. Σύγκριση

Η δεύτερη βασική αρχή προγραμματισμού είναι η σύγκριση. Στον προγραμματισμό μπορούν να συγκριθούν δυο ομοειδείς μεταβλητές για το ποια είναι μεγαλύτερη ή μικρότερη από μια άλλη.

$X = 5$

If $X > 4$ then κάνε κάτι

Η απλούστερη περίπτωση σύγκρισης όπου βλέπουμε μια μεταβλητή τύπου αριθμός αν είναι μεγαλύτερη από κάποιον απλό αριθμό κι έτσι αποφασίζουμε αν θα γίνει κάτι.

$X = \text{'Αποθήκη'}$

If $X = \text{'Αγορές'}$ then κάνε κάτι

Αντίστοιχη περίπτωση σύγκρισης όπου πλέον συγκρίνονται δυο κείμενα (string) για την ομοιότητά τους.

2.4. Μεταφορά

Η μεταφορά είναι η δυνατότητα να μεταφέρεται ο χρήστης από ένα σημείο του προγράμματος σε ένα άλλο έτσι ώστε να εκτελούνται ανάλογα με τις συνθήκες που υπάρχουν οι εντολές που θέλει. Η μεταφορά δηλαδή εξαρτάται από τις συγκρίσεις που κάνουμε για να επιλέξει ποιες ομάδες εντολών θα εκτελεστούν.

If $x < 5$ then

$Y = 2$

Else

$Y = 4$

End if

Εδώ έχουμε ένα απλό παράδειγμα όπου η μεταφορά λειτουργεί με βάση την εντολή σύγκρισης if και ανάλογα με το αποτέλεσμα της σύγκρισης δίνει την ανάλογη τιμή στο Y.

Call Υποπρόγραμμα1

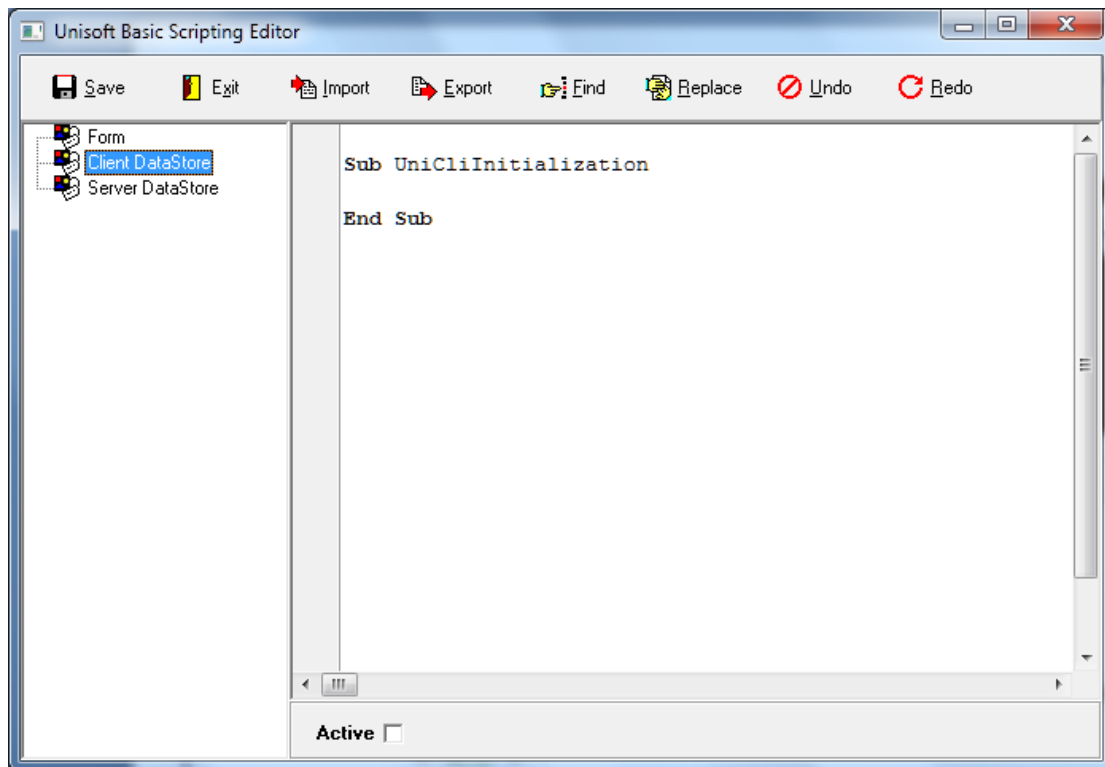
Εδώ μεταφέρει ο χρήστης την ροή του προγράμματος σε ένα υποπρόγραμμα το οποίο αφού ολοκληρωθεί θα επαναφέρει την λειτουργία του προγράμματος στην εντολή αμέσως μετά από την εντολή που call.

2.5. Αρχές Συγγραφής Προγράμματος

Κάθε πρόγραμμα είναι μια σειρά βασικών εντολών οι οποίες εκτελούνται με τη σειρά με την οποία γράφονται.

Ο υπολογιστής αρχίζει από την πρώτη γραμμή του προγράμματος και τελειώνει στην τελευταία εντολή. Για να γίνει πιο δυναμικός ο τρόπος προγραμματισμού, έχει υιοθετηθεί η φιλοσοφία των υποπρογραμμάτων (υπό-ρουτίνες και συναρτήσεις).

Στη περίπτωση της VBScript στο περιβάλλον του Atlantis το πρόγραμμα εκτελείται πάντα με την υπό-ρουτίνα UniCliInitialization, όπου μέσα ο χρήστης βάζει τις εντολές που θέλει και τελειώνει με το τέλος της συγκεκριμένης υπό-ρουτίνας.



Κάθε γραμμή της UniScript είναι και μια ολοκληρωμένη εντολή π.χ.

```
X=5
```

```
IF X=7 THEN CALL DOSOMETHING
```

Υπάρχει η δυνατότητα να έχουμε δύο εντολές σε μια γραμμή με το διαχωριστικό : (άνω κάτω τελεία) π.χ.

```
X=5 : Y=7
```

Αν μια γραμμή δεν χωρά την εντολή που θέλει ο χρήστης να γράψει, μπορεί να συνεχίσει στην επόμενη γραμμή με το σύμβολο _ (underscore).

```
Myvar = "This is a test text which continue to" & _  
"next line"
```

Υπάρχουν βέβαια σύνθετες εντολές που αποτελούνται από πολλές γραμμές κώδικα και τελειώνουν με μια εντολή τύπου end κάτι.

```
IF X=7
```

```

CALL DOSOMETHING
...
END IF

DO UNTIL X=7
    CALL DO SOMETHING
...
END LOOP

```

Όμοια δουλεύουν και οι υπό-ρουτίνες με αρχή το *sub* όνομα και τέλος το *end sub*.

2.6. Ρουτίνες και συναρτήσεις

Οι υπό-ρουτίνες (subroutines) είναι διαδικασίες που βρίσκονται κάπου μέσα στο πρόγραμμα και εκτελούν μια εργασία που θέλει ο χρήστης. Οι συναρτήσεις είναι υπό-ρουτίνες που επιστρέφουν μια τιμή π.χ.

```

Sub something
    X=3
    Y=5
    Z=area(x,y)
End sub

Function area(x,y)
    Area=x*y
End function

```

Η παραπάνω συνάρτηση area υπολογίζει το γινόμενο $x*y$ και μας δίνει το αποτέλεσμα. Κάθε συνάρτηση ή υπό-ρουτίνα μπορεί να δέχεται πολλές παραμέτρους.

Παράδειγμα 2 υπολογισμός ° Celsius από Fahrenheit.

```

Sub ConvertTemp()
    temp = InputBox("Please enter the temperature in degrees
F.", 1)
    MsgBox "The temperature is " & Celsius(temp) & " degrees
C."
End Sub

Function Celsius(fDegrees)
    Celsius = (fDegrees - 32) * 5 / 9
End Function

```

2.7. Τύποι δεδομένων

Κάθε μεταβλητή στην UniScript είναι τύπου VARIANT. Αυτό σημαίνει ότι μπορεί να είναι οποιασδήποτε μορφής.

Οι μορφές αυτές είναι:

Subtype	Description
Empty	VARIANT is uninitialized. Value is 0 for numeric variables or a zero-length string ("") for string variables.
Null	VARIANT intentionally contains no valid data.
Boolean	Contains either True or False .
Byte	Contains integer in the range 0 to 255.
Integer	Contains integer in the range -32,768 to 32,767.
Currency	-922,337,203,685,477.5808 to 922,337,203,685,477.5807.
Long	Contains integer in the range -2,147,483,648 to 2,147,483,647.
Single	Contains a single-precision, floating-point number in the range - 3.402823E38 to -1.401298E-45 for negative values; 1.401298E-45 to 3.402823E38 for positive values.
Double	Contains a double-precision, floating-point number in the range - 1.79769313486232E308 to -4.94065645841247E-324 for negative values; 4.94065645841247E-324 to 1.79769313486232E308 for positive values.
Date (Time)	Contains a number that represents a date between January 1, 100 to December 31, 9999.
String	Contains a variable-length string that can be up to approximately 2 billion characters in length.
Object	Contains an object.
Error	Contains an error number.

Μπορούμε λοιπόν να πούμε:

```
X=5
```

```
X="05-20-2000"
```

```
X="Κάτι"
```

```
X= True
```

```
Set X=CreateObject ("Scripting.FileSystemObject") κ.τ.λ.
```


Όταν η μεταβλητή είναι απλή η καταχώρηση γίνεται με την εντολή Let ή χωρίς καμία εντολή

```
Let x=5 :: x=5
```

Οι παραπάνω εντολές είναι ομότιμες.

Όταν η μεταβλητή είναι αντικείμενο η καταχώρηση γίνεται με την εντολή SET.

```
Set X=CreateObject("Scripting.FileSystemObject")
```

2.8. Δομημένος προγραμματισμός

Η UniScript έχει δύο ομάδες εντολών για δομή στα προγράμματα. Η πρώτη είναι η ομάδα ελέγχου και η δεύτερη η ομάδα συνθηκών.

Εντολές ελέγχου:

```
If {συνθήκη} then {εντολές} else {εντολές}
```

```
If {συνθήκη} then
```

```
    {εντολές}
```

```
else
```

```
    {εντολές}
```

```
end if
```

```
if {συνθήκη}
```

```
    elseif {εντολές}
```

```
    elseif {εντολές}
```

```
    ...
```

```
    else
```

```
endif
```

```
Select case {μεταβλητή}
```

```
    case {συνθήκη1} {εντολές}
```

```
    case {συνθήκη2} {εντολές}
```

```
    ...
```

```
end select
```

Εντολές συνθηκών

```
Do while {συνθήκη}
```

```
    {εντολές}
```

```
loop
```

```
Do
```

```

{ εντολές }
loop while { συνθήκη }

do until { συνθήκη }
  { εντολές }
loop

do
  { εντολές }
loop until { συνθήκη }

while { συνθήκη }
  { εντολές }
wend

for {μετρητής (I=1 to 10)}
  { εντολές }
next

for each {item} in {group}
  { εντολές }
next

Sub cmdChange_OnClick
  Dim d                                'Create a variable
  Set d = CreateObject("Scripting.Dictionary")
  d.Add "0", "Athens"                  'Add some keys and items
  d.Add "1", "Belgrade"
  d.Add "2", "Cairo"

  For Each I in d
    Document.frmForm.Elements(I).Value = D.Item(I)
  Next
End Sub

```

2.9. Objects και Class

Object είναι η υπόσταση στη μνήμη μιας Class.

Class είναι το σύνολο μεταβλητών υπό-ρουτινών και συναρτήσεων που κάτω από μια κοινή ονομασία δίνονται στο περιβάλλον λειτουργίας της UniScript π.χ. IUniScriptObject.

Το IUniScriptObject αποτελείται από μία συνάρτηση την DoError και τρία Object.

Ο ορισμός του είναι μια Class, η χρήση του σε μια μεταβλητή γίνεται σαν χρήση Object.

Η διαφορά πρακτικά Class και Object είναι ότι την Class την ορίζει ο χρήστης μια φορά και δημιουργεί όσα object τύπου class θέλει.

Στην UniScript δεν μπορεί να ορίσει Class. Μπορεί όμως έτοιμες Class να τις χρησιμοποιεί (σαν Object's). π.χ.

```
IUniScriptFieldObject
```

```
Property Value: OleVariant read Get_Value write Set_value;  
Property OnChange: WideString write Set_OnChange;  
Property OnValidate: WideString write Set_OnValidate;  
end;
```

Παραπάνω υπάρχει η δήλωση μιας Class ενός πεδίου όπως αυτή ορίζεται στο Delphi και παρέχεται για χρήση στη UniScript. Όταν ο χρήστης θέλει να χρησιμοποιήσει ένα πεδίο ενός πίνακα το μόνο που έχει να κάνει είναι:

```
Set tblCustomer=IClientDM.ICliTable("CUSTOMER")  
Set FldCusName=tblCustomer.IField("NAME")
```

Η μεταβλητή tblCustomer γίνεται τύπου Object με τις ιδιότητες πίνακα (class IUniscrptTableObject), και η μεταβλητή τύπου FldCusName γίνεται Object τύπου (class) IUniScriptFieldObject.

Κάθε property, function και subroutine (procedure) ενός object χρησιμοποιείται με την λογική object.property βάζοντας δηλαδή μια τελεία ανάμεσα στο object και το στοιχείο του που επιθυμεί ο χρήστης π.χ.

```
FldCusName.value  
FldCusName.onchange  
FldCusName.onvalidate
```

Τα property είναι οι μεταβλητές και τα event κάθε αντικείμενου.

Το FldCusName έχει τις ιδιότητες value, onchange και onvalidate. Το value είναι η μεταβλητή που έχει την τιμή του πεδίου.

Έχει οριστεί με ιδιότητες Get και Set που σημαίνει ότι ο χρήστης μπορεί να διαβάσει και να γράφει σε αυτή τη μεταβλητή. (Get διαβάζω Set γράφω).

Τα events ορίζονται (με απλοϊκό τρόπο) ως τα property που αρχίζουν με τη λέξη on. Έχουν μόνο set ιδιότητα που σημαίνει ότι μπορούν να ενεργοποιηθούν.

Το πρώτο property κάθε Class είναι το default. Το default property αποκτά τη δυνατότητα να απευθύνεται ο χρήστης σ' αυτό όταν δίνει τιμή σε ένα αντικείμενο χωρίς την εντολή Set.

Έτσι ενώ όρισε παραπάνω την FldCusName σαν object μπορεί να πει FldCusName=5 που όμως πρακτικά σημαίνει FldCusName.value=5.

2.10. Έλεγχος λαθών

Η VBScript έχει μόνο μια εντολή ελέγχου λαθών την:

```
On error resume next
```

η οποία σημαίνει ότι αν κάποιο λάθος προκύψει από την εκτέλεση του κώδικα, αυτόματα ο χρήστης θα μεταφερθεί στην επόμενη γραμμή εντολής.

Έτσι αν επιθυμεί ο χρήστης να εισάγει έναν αριθμό σε κάποια φάση του προγράμματος η διαδικασία θα πρέπει να είναι:

```
Function getnum
  Dim x as real
  x= inputbox("Δώστε μια τιμή")
  Do until IsReal(x)
    x= inputbox("Δώστε μια σωστή τιμή ")
  loop
  getnum=x
end function
```

3. Αντικείμενα της VBScript

3.1. Dictionary object

Είναι ένα αντικείμενο που κρατά ομάδες δεδομένων (κλειδί και τιμή) στη μνήμη. Η βασική του ιδιότητα είναι να βοηθά την εντολή for ... each.

```
Dim d ' Create a variable.
Set d = CreateObject("Scripting.Dictionary")
d.Add "a", "Athens" ' Add some keys and items.
d.Add "b", "Belgrade"
d.Add "c", "Cairo"
...
```

Properties...

CompareMode: Παίρνει δύο τιμές. 0 (μηδέν) όταν συγκρίνει binary δεδομένα (αριθμοί, ημερομηνίες κτλ.) και 1 όταν συγκρίνει κείμενα (string).

Count: Ο αριθμός των δεδομένων που υπάρχουν στο αντικείμενο.

```
Function ShowKeys
  Dim a, d, i, s ' Create some variables.
  Set d = CreateObject("Scripting.Dictionary")
  d.Add "a", "Athens" ' Add some keys and items.
  d.Add "b", "Belgrade"
  d.Add "c", "Cairo"
```

```

a = d.Keys          ' Get the keys.
For i = 0 To d.Count -1 ' Iterate the array.
    s = s & a(i) & "<BR>" ' Create return string.
Next
ShowKeys = s
End Function

```

Item: Το property που αναφέρεται στην τιμή του τρέχοντος δεδομένου.

```

Function ItemDemo
    Dim d          ' Create some variables.
    Set d = CreateObject("Scripting.Dictionary")
    d.Add "a", "Athens" ' Add some keys and items.
    d.Add "b", "Belgrade"
    d.Add "c", "Cairo"
    ItemDemo = d.Item("c") ' Get the item.
End Function

```

Key: Το property που αναφέρεται σε κλειδί και επιλέγει δεδομένο ή αλλάζει την τιμή της μεταβλητής του.

```

Function DicDemo
    Dim d          ' Create some variables.
    Set d = CreateObject("Scripting.Dictionary")
    d.Add "a", "Athens" ' Add some keys and items.
    d.Add "b", "Belgrade"
    d.Add "c", "Cairo"
    d.Key("c") = "d" ' Set key for "c" to "d".
    DicDemo = d.Item("d") ' Return associate item.
End Function

```

Methods

Add: προσθέτει μία νέα τιμή στον πίνακα

Exist: Ελέγχει για την ύπαρξη μιας τιμής

Items: Επιστρέφει έναν πίνακα με τα περιεχόμενα του object

Keys: Επιστρέφει έναν πίνακα με τα κλειδιά του object

Remove: Αφαιρεί μια μεταβλητή

RemoveAll: Αφαιρεί όλα τα περιεχόμενα.

3.2. Drive object

Το object Drive δίνει ένα σύνολο πληροφοριών σχετικά με την ενεργή συσκευή αποθήκευσης (σκληρός δίσκος, δισκέτα, CD, mapped drive στο δίκτυο κτλ.)

Properties...

AvailableSpace: Διαθέσιμος χώρος για τον χρήστη σε drive δικτύου.

DriveLetter: Το γράμμα με το οποίο απεικονίζεται η συσκευή.

DriveType: Το είδος της συσκευής (Δισκέτα, σκληρός κτλ.)

FileSystem: Το σύστημα αρχείων (Fat16, Fat32, NTFS κτλ.)

FreeSpace: Ο ελεύθερος χώρος στη συσκευή.

IsReady: Αν είναι ενεργή η συσκευή (CD, remonables).

Path: Ο ενεργός κατάλογος (path) της συσκευής.

RootFolder: Ο κύριος κατάλογος της συσκευής (έχει νόημα σε δίκτυο).

SerialNumber: Ο αριθμός σειράς της συσκευής.

ShareName: Το όνομα αντιστοίχισης στο δίκτυο.

TotalNumber: Το μέγεθος σε bytes της συσκευής.

VolumeName: Το όνομα (που δίνεται στο format) της συσκευής.

Το object Drive είναι μόνο πληροφοριακό και δεν έχει Methods.

3.3. Drives Collection

Είναι ένα αντικείμενο μορφής συλλογή το οποίο περιέχει objects τύπου drive για χρήση στην εντολή for .. each.

```
Function ShowDriveList
```

```
    Dim fso, d, dc, s, n
    Set fso = CreateObject("Scripting.FileSystemObject")
    Set dc = fso.Drives
```

```
    For Each d in dc
        n = ""
        s = s & d.DriveLetter & " - "
        If d.DriveType = Remote Then
            n = d.ShareName
        ElseIf d.IsReady Then
            n = d.VolumeName
        End If
        s = s & n & "<BR>"
    Next
```

```
    ShowDriveList = s
End Function
```

Το παραπάνω παράδειγμα δίνει έναν πίνακα με τα διαθέσιμα drive.

3.4. File object

Το βασικότερο object για χρήση αρχείων. Μας δίνει πρόσβαση σε ένα αρχείο, είτε να το δημιουργήσουμε είτε να το διαγράψουμε είτε γενικά να κάνουμε αλλαγές σ' αυτό.

Properties...

Attributes: Κατάσταση του αρχείου (κανονικό, κρυφό, συστήματος κτλ).

DateCreated: Ημερομηνία δημιουργίας αρχείου (NT μόνο).

DateLastAccessed: Ημερομηνία τελευταίας πρόσβασης (NT μόνο).

DateLastModified: Ημερομηνία τελευταίας αλλαγής.

Drive: Συσκευή στην οποία βρίσκεται το αρχείο (τύπου Drive Object)

Name: Όνομα αρχείου.

ParentFolder: Κατάλογος στον οποίο βρίσκεται το αρχείο

Path: Διαδρομή καταλόγων στην οποία βρίσκεται το αρχείο.

ShorName: Όνομα αρχείου με απεικόνιση 8.3

ShortPath: Όνομα διαδρομής με απεικόνιση 8.3

Size: Μέγεθος αρχείου

Type: Τύπος αρχείου (αρχείο ή κατάλογος).

Methods...

Copy: Αντιγράφει το αρχείο.

```
Set MyFile = fso.GetFile("c:\testfile.txt")
```

```
MyFile.Copy ("c:\windows\desktop\test2.txt")
```

Delete: Διαγράφει το αρχείο.

Move: Μεταφέρει το αρχείο σε άλλο κατάλογο.

OpenAsStream ([iomode, [format]]): Ανοίγει ένα text αρχείο για διάβασμα, εγγραφή ή προσθήκη.

Το IOMode παίρνει τις τιμές :

Constant	Value	Description
ForReading	1	Open a file for reading only. You can't write to this file.
ForWriting	2	Open a file for writing. If a file with the same name exists, its previous contents are overwritten.
ForAppending	8	Open a file and write to the end of the file.

Το format παίρνει τις τιμές :

Constant	Value	Description
TristateUseDefault	-2	Opens the file using the system default.
TristateTrue	-1	Opens the file as Unicode.
TristateFalse	0	Opens the file as ASCII.

```
Function TextStreamTest
```

```
Const ForReading = 1, ForWriting = 2, ForAppending = 8
```

```

Const TristateUseDefault = -2, TristateTrue = -1
Const TristateFalse = 0
Dim fso, f, ts
Set fso = CreateObject("Scripting.FileSystemObject")
fso.CreateTextFile "test1.txt" ' Create a file.
Set f = fso.GetFile("test1.txt")
Set ts = f.OpenAsTextStream(ForWriting, _
TristateUseDefault)
ts.Write "Hello World"
ts.Close
Set ts = f.OpenAsTextStream(ForReading, _
TristateUseDefault)
TextStreamTest = ts.ReadLine
ts.Close
End Function

```

Η παραπάνω συνάρτηση ανοίγει ένα αρχείο test1.txt για εγγραφή και για προσθήκη. Η method OpenAsTextStream του object FileSystemObject λειτουργεί όπως και η OpenAsStream του object File.

3.5. Files Collection

Το Files Collection είναι μια συλλογή από αντικείμενα τύπου File.

3.6. FileSystemObject Object

Το FileSystemObject είναι το βασικό αντικείμενο αλληλεπίδρασης με το λειτουργικό σύστημα.

Properties...

Drives: Collection τύπου Drives με τις διαθέσιμες συσκευές του συστήματος.

Methods...

BuildPath: Προσθέτει μια επιπλέον διαδρομή στο τρέχον Path.

CopyFile: Αντιγράφει το τρέχον επιλεγμένο αρχείο.

CopyFolder: Αντιγράφει έναν κατάλογο από μια θέση σε μια άλλη.

CreateFolder: Δημιουργεί έναν κατάλογο.

CreateTextFile: Δημιουργεί ένα αρχείο κειμένου. (όπως στο File Object)

DeleteFile: Διαγράφει το τρέχον επιλεγμένο αρχείο.

DeleteFolder: Διαγράφει έναν κατάλογο.

DriveExist: Ελέγχει την ύπαρξη μιας συσκευής αποθήκευσης.

FileExist: Ελέγχει την ύπαρξη ενός αρχείου.

FolderExist: Ελέγχει την ύπαρξη ενός καταλόγου.

GetAbsolutePathName: Επιστρέφει την πλήρη διαδρομή ενός αρχείου (για χρήση σε σίκτυο).

GetBaseName: Επιστρέφει το όνομα του τρέχοντος αρχείου χωρίς την κατάληξη του.

GetDrive: Επιστρέφει ένα object τύπου Drive.

GetDriveName: Επιστρέφει το όνομα της συσκευής που επιλέξαμε.

GetExtentionName: Επιστρέφει την κατάληξη του ονόματος του τρέχοντος αρχείου.

GetFile: Επιλέγει ένα αρχείο.

GetFileName: Επιστρέφει το όνομα του αρχείου μαζί με τη διαδρομή του, χωρίς το όνομα της συσκευής αποθήκευσης.

GetFolder: Επιλέγει έναν κατάλογο ως ενεργό.

GetParentFolderName: Επιστρέφει τη διαδρομή του τρέχοντος ενεργού καταλόγου.

GetSpecialFolder: Επιστρέφει το Special Folder που ζητάει ο χρήστης.

GetTempName: Επιστρέφει ένα τυχαίο όνομα αρχείου (το οποίο δεν υπάρχει στον υπολογιστή).

MoveFile: Μετακινεί ένα αρχείο.

MoveFolder: Μετακινεί έναν κατάλογο.

OpenTextFile: Ανοίγει ένα αρχείο κειμένου.

3.7. Folder Object

Το Folder Object είναι το βασικό αντικείμενο διαχείρισης καταλόγων.

Properties...

Attributes: Γράφει σε ή διαβάζει τα χαρακτηριστικά ενός καταλόγου.

DateCreated: Ημερομηνία δημιουργίας καταλόγου.

DateLastAccessed: Ημερομηνία τελευταίας πρόσβασης στον κατάλογο.

DateLastModified: Ημερομηνία τελευταίας αλλαγής στον κατάλογο.

Drive: Συσκευή στην οποία βρίσκεται ο κατάλογος.

Files: Συλλογή με τα αρχεία που περιέχει ο κατάλογος.

IsRootFolder: True αν ο κατάλογος είναι πρώτος στη δομή.

Name: Όνομα καταλόγου.

ParentFolder: Όνομα του καταλόγου στον οποίο βρίσκεται ο τρέχων κατάλογος.

Path: Διαδρομή στην οποία βρίσκεται ο τρέχων κατάλογος.

ShortName: Το όνομα του καταλόγου σε μοφή 8.3

ShortFolder: Επιστρέφει τη διαδρομή του καταλόγου σε μορφή 8.3

Size: Επιστρέφει το άθροισμα των μεγεθών των αρχείων που βρίσκονται στον κατάλογο, σε bytes.

SubFolders: Επιστρέφει μια συλλογή από καταλόγους με τους υποκαταλόγους που υπάρχουν στον τρέχων κατάλογο.

Type: Επιστρέφει τον τύπο του καταλόγου (ο τύπος εξαρτάται από την κατάληξη).

Methods...

Copy: Αντιγράφει έναν κατάλογο.

Delete: Διαγράφει τον τρέχοντα κατάλογο.

Move: Μετακινεί τον τρέχοντα κατάλογο.

CreateTextFile: Δημιουργεί ένα αρχείο κειμένου.

3.8. Folders Collection

Μια συλλογή από object τύπου Folder

3.9. TextStream Object

Το TextStream είναι το αντικείμενο που διαχειρίζεται αρχεία κειμένου.

Properties...

AtEndOfLine: Επιστρέφει true αν ο pointer στο αρχείο βρίσκεται στο τέλος της γραμμής, διαφορετικά επιστρέφει false.

AtEndOfStream: Επιστρέφει true αν ο pointer είναι στο τέλος του αρχείου.

Column: Επιστρέφει την στήλη της τρέχουσας γραμμής στην οποία είναι ο pointer.

Line: Επιστρέφει την τρέχουσα γραμμή στην οποία βρίσκεται ο pointer.

Methods...

Close: Κλείνει το τρέχον αρχείο.

Read: Διαβάζει έναν αριθμό χαρακτήρων από το αρχείο.

ReadAll: Διαβάζει όλο το αρχείο κειμένου.

ReadLine: Διαβάζει μια ολόκληρη γραμμή.

Skip: Μεταφέρει τον pointer σε οριζόμενο πλήθος χαρακτήρων μετά την τρέχουσα θέση.

SkipLine: Μετακινεί τον pointer μια γραμμή παρακάτω.

Write: Γράφει ένα συγκεκριμένο κείμενο (string) στο αρχείο (χωρίς σημάδι τέλους γραμμής).

WriteLine: Γράφει ένα κείμενο (string) και κατόπιν αλλάζει γραμμή.

WriteBlankLines: Εισάγει στο αρχείο ένα σύνολο κενών γραμμών που θα ορίσουμε.

3.10. Παρατηρήσεις

Όλα τα αντικείμενα που έχει η VBScript δεν ενεργοποιούν event.

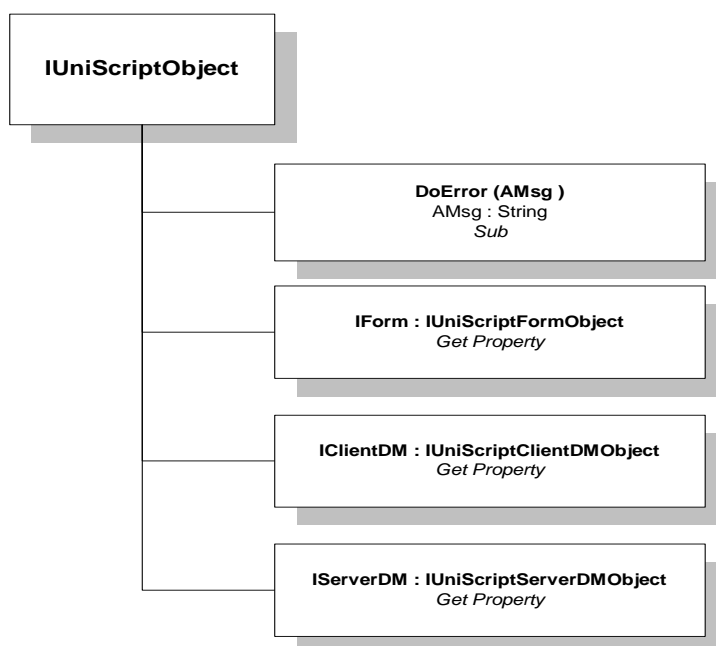
Έτσι, για οποιαδήποτε περίπτωση λάθους ή ιδιόμορφης λειτουργίας θα πρέπει ο χρήστης να έχει γράψει τον ανάλογο κώδικα.

π.χ. Δεν έχουμε event για το τέλος αρχείου οπότε θα πρέπει εμείς συνεχώς να ελέγχουμε το AtEndOfStream property.

4. Object Model της Altec VB Language

4.1. IUniScriptObject

Η παρουσίαση των Objects θα γίνει με κωδικοποίηση Delphi όπως αυτό χειρίζεται και απεικονίζει τα μοντέλα του στην επικοινωνία με τη VBScript



Το παραπάνω object είναι το πρώτο που δημιουργείται με την λειτουργία Script Basic Editor και χρησιμοποιεί την μεταβλητή uni. Παρέχει στον χρήστη τη δυνατότητα να χειριστεί τις τρεις βασικές καταστάσεις του εκάστοτε παραθύρου, τη Φόρμα, τον Client και τον Server του Atlantis.

Στην χρήση δεν θα χρειαστεί το παραπάνω object, αλλά το μόνο που χρειάζεται να γνωρίζει είναι ότι κατά την έναρξη της λειτουργίας της VBScript έχει ήδη ενεργό ένα Instance του παραπάνω object.

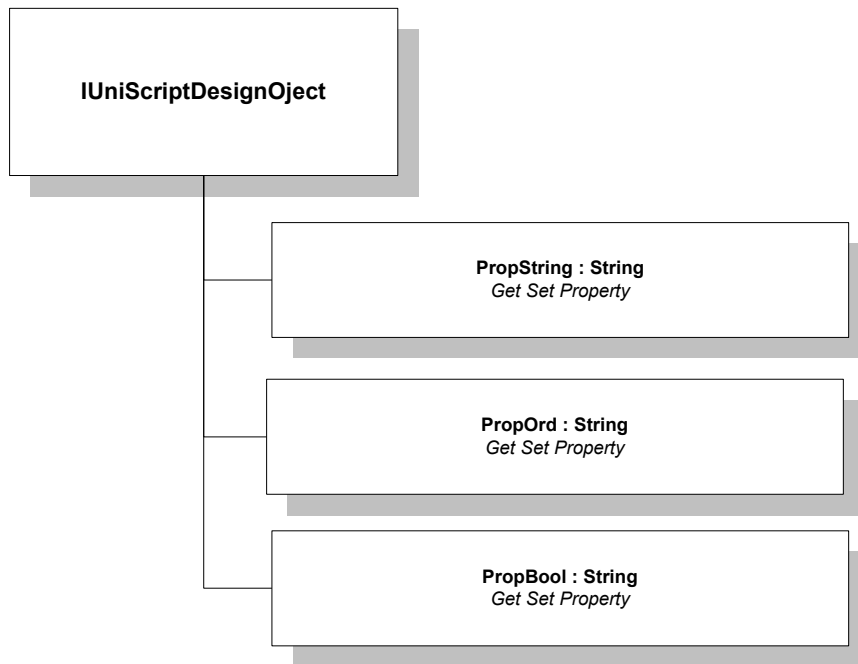
4.2. IUniscrptDesignObject

Το IUniscrptDesignObject είναι το object που ελέγχει τα Control της φόρμας.

Το PropString έχει το caption ενός Control που εμφανίζει η φόρμα.

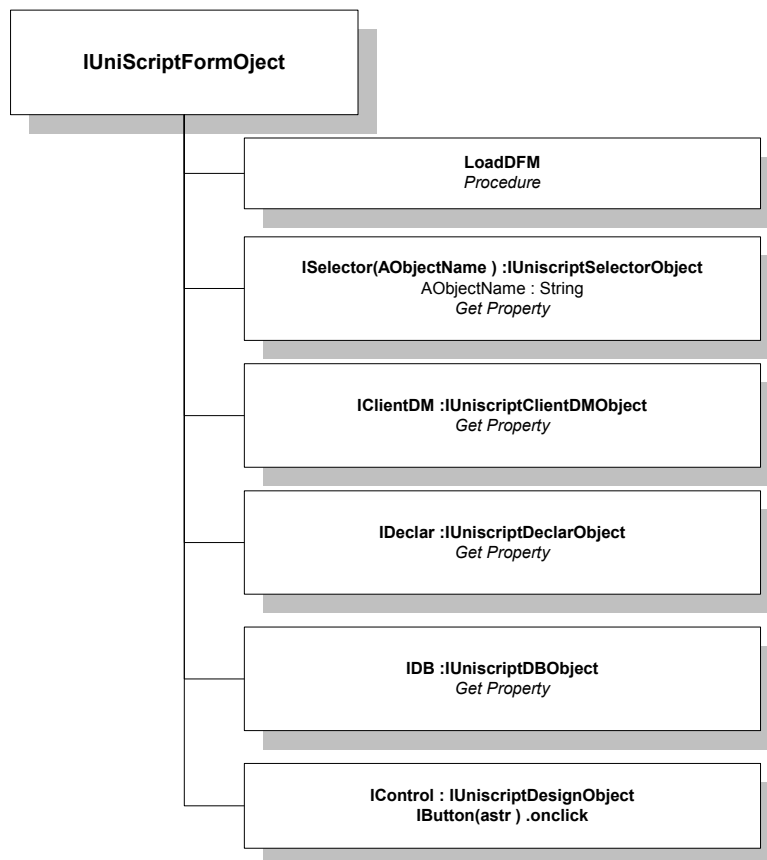
Το PropOrd δίνει τιμή σε ένα property.

Το PropBool δίνει True or False σε ένα property.



4.3. IUniscrptFormObject

Το IUniscrptFormObject είναι το βασικό object στην λειτουργία Form του αριστερού παραθύρου της UB Scripting Language. Το IUniscrptFormObject είναι απόγονος του IUniscrptDesignObject οπότε έχει και όλες τις ιδιότητες του.



Η procedure LoadDFM είναι ανενεργή...

Το Property ISelector δίνει ένα instance ενός selector Object.

Το Property IClientDM δίνει ένα instance του Client Object

Το Property IDeclar δίνει ένα instance του IUniscriptDeclarObject.

Τέλος το IDB δίνει ένα instance τύπου IUniscriptDBObject.

Η method iControl δίνει την δυνατότητα χρήσης του Design Object. Με το property PropOrd, ορίζονται τα περιεχόμενα των properties στα control κάθε φόρμας.

Π.χ. έστω ότι ο χρήστης θέλει να αλλάξει το μέγεθος του ορίου αξιογράφων πελάτη στη φόρμα τραπεζών....

Διαχείριση τραπεζών

Λίστα

Γενικά Υποκαταστήματα Λογαριασμοί

Στοιχεία ταυτότητας

Κωδικός Επωνυμία ALPHA

Κωδικός Γ.Λ. Κωδικός BIC

Παράμετροι εξόφλησης

Ημέρες Valeur

Προμήθεια %

Προεξοφλητικό επιτόκιο %

Προμήθεια ανά αξιόγραφο

Όριο αξιόγραφων ανά πελάτη

Νέα εγγραφή

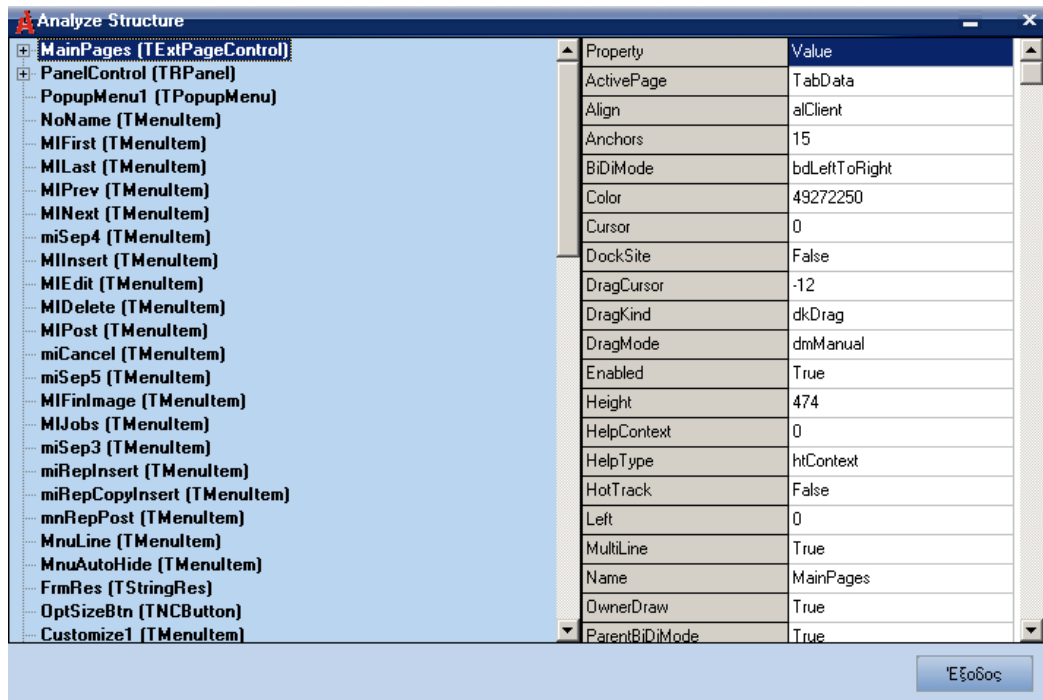
Διαγραφή

Καταχώριση

Ακύρωση

Ανανέωση

Ο χρήστης βρίσκει το όνομα του σχετικού control στο Analyze Structure της φόρμας... (με τον συνδυασμό Alt-Ctrl-Shift-Z)



και χρησιμοποιούμε την εντολή

```
form.icontrol("RDBEdit6").propord("Width")=400
```

για να αλλάξει το πλάτος του control από 80 σε 400.

Τα properties που μπορεί ο χρήστης να αλλάξει είναι αυτά που περιγράφονται στο δεξί παράθυρο.

Με την ίδια λογική δουλεύει και το iButton. Επιλέγει από το Analyze Structure το Button στο οποίο θέλει να ενεργοποιήσει το onclick event και ορίζει μια subroutine η οποία θα χρησιμοποιηθεί π.χ.

```
iButton("NewRecBtn").onclick="DoMySub"
```

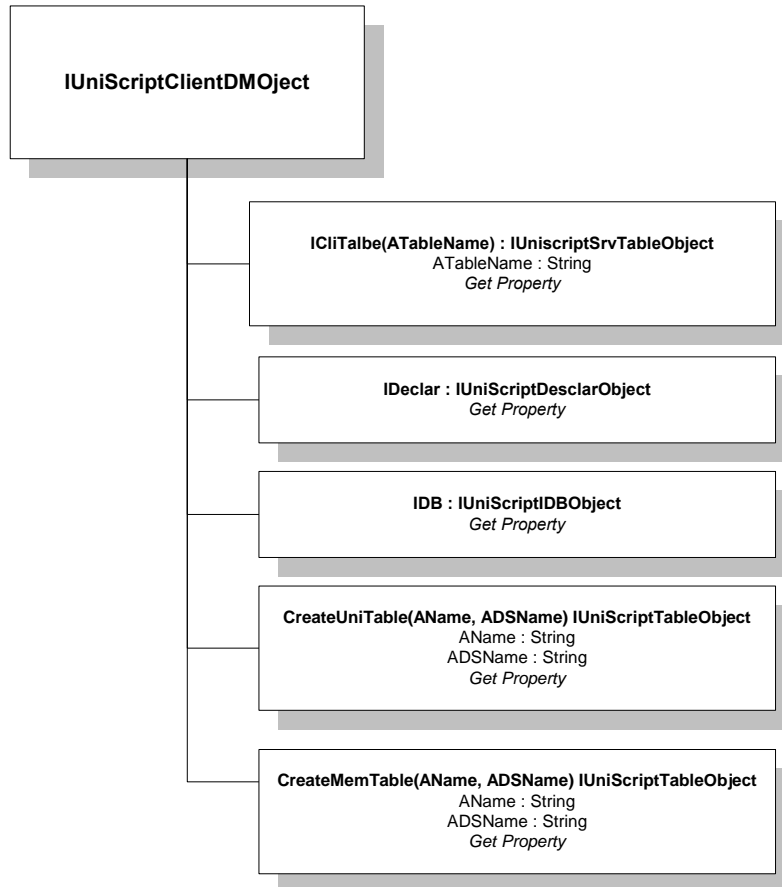
Τα πιο κοινά buttons είναι:

NewRecBtn, DeletBtn, RefrBtn στον Browser και

FirstBtn, LastBtn, PrevBtn, NextBtn, RefrDataBtn, FindDataBtn, PostBtn, CancelBtn, ExitBtn στην Data Page.

4.4. IUniScriptClientDMObject

Το IUniScriptClientDMObject είναι το βασικό object στην λειτουργία Client του αριστερού παραθύρου της UB Scripting Language.



Το property ICliTable δίνει ένα instance του πίνακα που ο χρήστης ορίζει με τη μεταβλητή (τύπου string) AObject.

Το Property IDeclar δίνει ένα instance του IUniscriptDeclarObject.

Το IDB δίνει ένα instance τύπου IUniscriptDBObject.

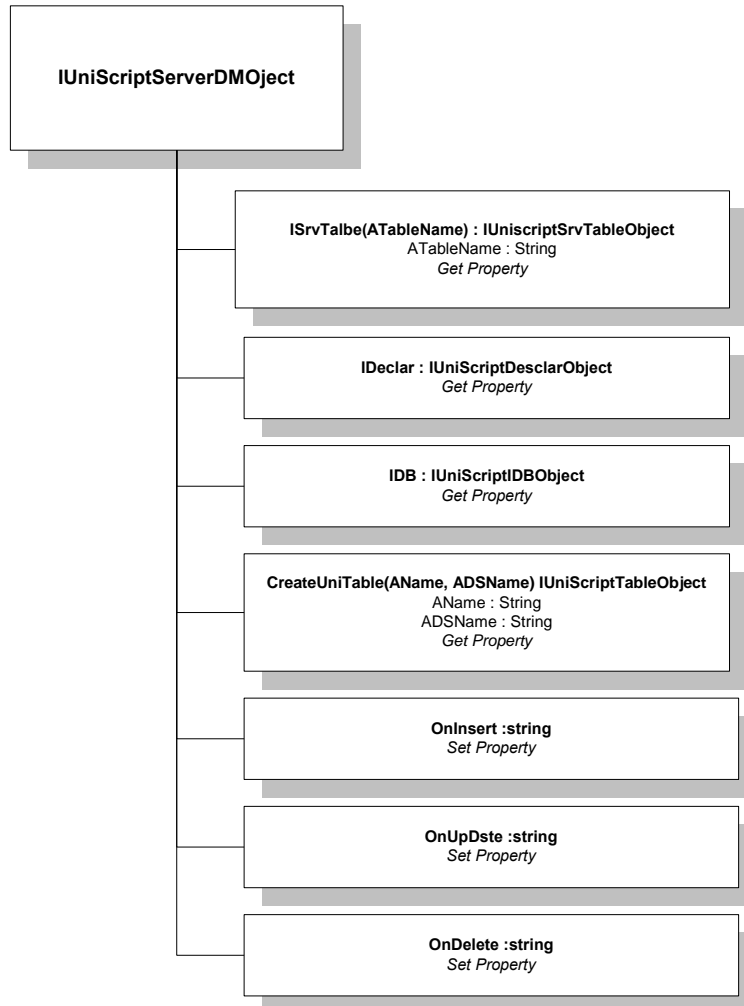
Το Property CreateUniTable διαβάζει τον πίνακα AName από το DSN ADSName.

Το Property CreateMemTable διαβάζει στην μνήμη τον πίνακα AName στο DSN ADSName.

Η βασική χρήση του IClientDM object είναι ο έλεγχος των δεδομένων που εισάγονται και κάποιες πιθανές επιπλέον ενημερώσεις σε δεδομένα της Class που είναι ενεργή στην φόρμα.

4.5. IUniScriptServerDMObject

Το IUniScriptServerDMObject είναι το βασικό object στην λειτουργία Client του αριστερού παραθύρου της UB Scripting Language.



Το Property ISrvTable διαβάζει τον πίνακα ATableName.

Το Property IDeclar δίνει ένα instance του IUniscriptDeclarObject.

Το IDB = δίνει ένα instance τύπου IUniscriptDBObject.

Το Property CreateUniTable διαβάζει τον πίνακα AName από το DSN ADSName.

Τα events OnInsert, OnUpDate και OnDelete ενεργοποιούνται πριν από την αντίστοιχη λειτουργία του Server αλλά αφού έχουν ολοκληρωθεί όλοι οι έλεγχοι στον Client και στον Application Server του Atlantis.

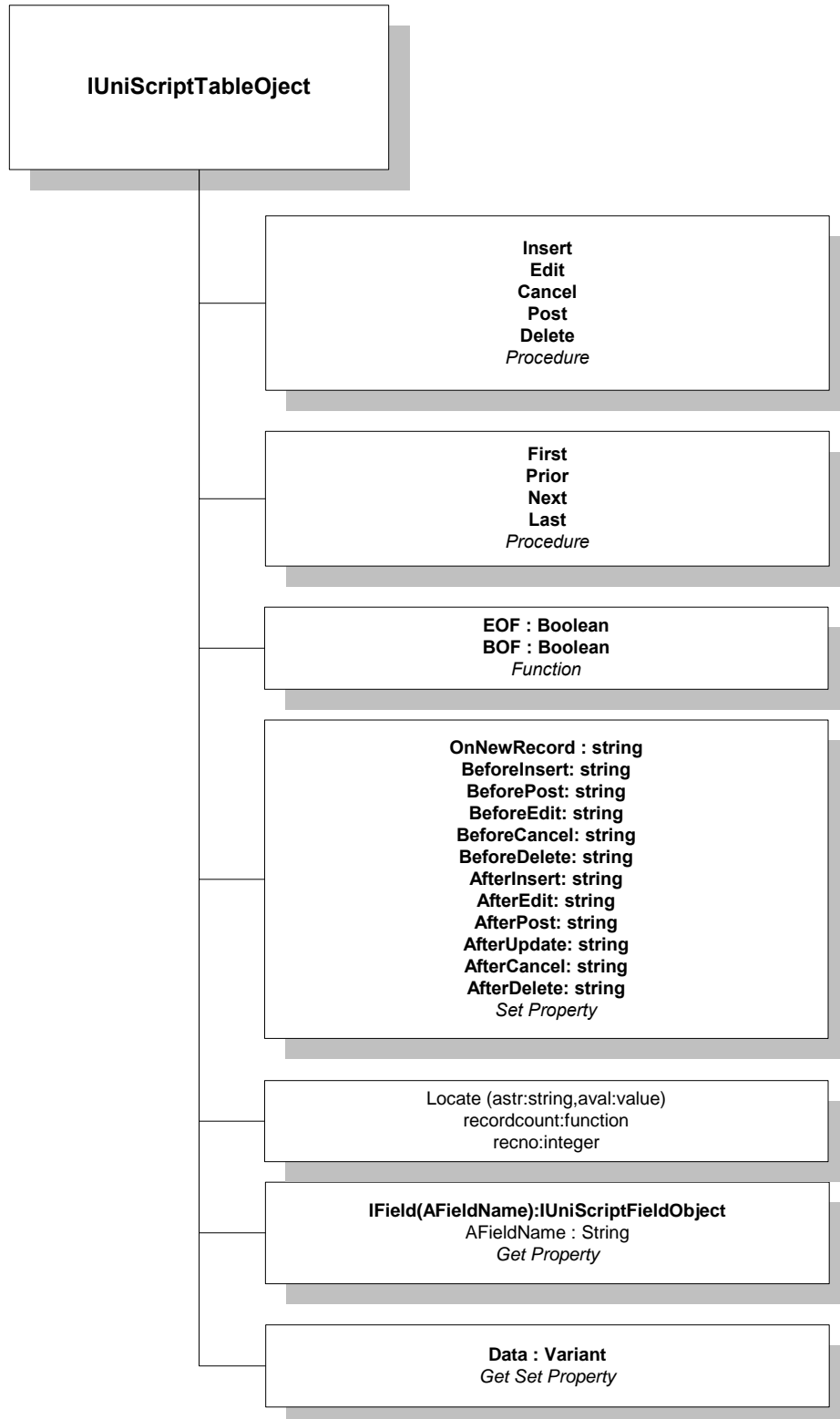
Έτσι πρακτικά αν ενεργοποιήσουμε κάποιο event αυτό θα ενεργοποιηθεί πραγματικά την στιγμή που θα εκτελέσει την ανάλογη εντολή ο Data Base Server.

Αν για παράδειγμα εκτελούμε μια εντολή διαγραφής στους πελάτες και ο πελάτης έχει κινήσεις και δεν μπορεί να διαγραφεί, δεν θα ενεργοποιηθεί το event onDelete, θα έχει σταματήσει πιο πριν η διαδικασία σε επίπεδο Atlantis και όχι βάσης.

Το IServerDM object χρησιμοποιείται για να ενημερώνει οποιοδήποτε table είναι απαραίτητο να ενημερωθεί σε διαδικασίες επιπέδου καταχώρησης π.χ. ενημέρωση καρτέλας και βοηθητικών στοιχείων.

4.6. IUniscriptTableObject

Το αντικείμενο αυτό χειρίζεται ένα recordset (Table) και δίνει τις σχετικές με αυτό λειτουργίες και event.



Το παραπάνω object δεν χρησιμοποιείται αυτόνομα αλλά αποκτά instance όταν το ορίζουμε μέσα από το IClientDM π.χ.

```
Set tblCustomer=IClientDM.ICliTable("CUSTOMER")
```

Το object IClientDM.ICliTable είναι τύπου IUniscriptTableObject.

Με την παραπάνω εντολή της VBScript η μεταβλητή tblCustomer παίρνει το recordset customer, όπως αυτό είναι ενεργό εκείνη τη στιγμή στο Atlantis. Αυτό σημαίνει ότι ο cursor θα βρίσκεται στην τρέχουσα εγγραφή που έχει ο χρήστης στην οθόνη του εκείνη τη στιγμή. Αν ο συγκεκριμένος πίνακας δεν είναι ενεργός στην Class της Φόρμας θα πρέπει ο χρήστης να διαβάσει τα δεδομένα με την IDB.QueryDATA. Την θέση του cursor σε αυτή την περίπτωση την ορίζουμε με τις εντολές μετακίνησης First, Last, Next και Prior.

Οι **εντολές μετακίνησης** μεταφέρουν τον Cursor ως εξής:

- **First:** ο cursor μετακινείται στην πρώτη εγγραφή του πίνακα
- **Next:** ο cursor μετακινείται στην επόμενη εγγραφή του πίνακα, απ' αυτή που βρισκόμαστε.
- **Prior:** ο cursor μετακινείται στην προηγούμενη εγγραφή και
- **Last:** ο cursor μετακινείται στην τελευταία εγγραφή του πίνακα. πχ.
- **TblCustomer.next:** μεταφέρει τον cursor στην πρώτη εγγραφή του πίνακα customer.

Οι συναρτήσεις **EOF** και **BOF** δείχνουν τη θέση μέσα στον πίνακα που βρίσκεται ο χρήστης. Ενεργοποιούνται από τις **Procedures Next** και **Prior** αντίστοιχα και έχουν την τιμή:

- **false** όταν η επόμενη (ή προηγούμενη) εγγραφή δεν είναι η τελευταία,
- **true** όταν η επόμενη (ή προηγούμενη) εγγραφή είναι η τελευταία (πρώτη)

Π.χ. αν ο χρήστης βρίσκεται στην δεύτερη εγγραφή με την εντολή **Prior** η **BOF** θα πάρει την τιμή **False** και ο cursor θα μεταφερθεί στην πρώτη εγγραφή.

Αν ξαναεκτελέσουμε την **Prior** cursor θα παραμείνει στην πρώτη εγγραφή και η **BOF** θα πάρει την τιμή **True**.

Επόμενη ομάδα εντολών είναι οι εντολές επεξεργασίας οι οποίες διορθώνουν, εισάγουν και διαγράφουν εγγραφές (record) στην μνήμη

(Δεν μπορούμε να αποθηκεύσουμε στη βάση τα data):

- **Insert:** εισάγει μια κενή εγγραφή για να προσθέσουμε τιμές
- **Edit:** διορθώνει την τρέχουσα εγγραφή (δεν μπορούμε να αλλάξουμε το βασικό κλειδί της εγγραφής)
- **Delete:** διαγράφει την τρέχουσα εγγραφή (Δεν διαγράφεται αν είναι συνδεδεμένη με άλλον πίνακα Foreign keys και Constrains.)

Οι παραπάνω εντολές εκτελούνται αν κατόπιν δώσουμε την εντολή **Post** ή ακυρώνονται με την εντολή **Cancel** (πάντα στην μνήμη).

Στο IUniscriptTableObject υπάρχουν 11 event για να μπορεί ο χρήστης να ελέγχει διαδικασίες πριν ή μετά την μεταβολή μιας εγγραφής.

Το event OnNewRecord ενεργοποιείται όταν εκτελείται μια διαδικασία εισαγωγής εγγραφής σε πίνακα.

Τα BeforeInsert και AfterInsert ενεργοποιούνται πριν και μετά την εκτέλεση μιας εντολής Insert.

Τα BeforeCancel και AfterCancel ενεργοποιούνται πριν και μετά την εκτέλεση μιας εντολής Cancel.

Τα BeforeEdit και AfterEdit ενεργοποιούνται πριν και μετά την εκτέλεση μιας εντολής Edit.

Τα BeforeDelete και AfterDelete ενεργοποιούνται πριν και μετά την εκτέλεση μιας εντολής Delete.

Τα BeforePost και AfterPost ενεργοποιούνται πριν και μετά την εκτέλεση μιας εντολής Post.

Η function locate ψάχνει στον πίνακα μια εγγραφή βάσει του πεδίου (ή των πεδίων με διαχωρισμό με το ;) astr και με τιμή (ή τιμές σε μια variant array) anvar και επιστρέφει true αν βρει την εγγραφή ή false αν δεν τη βρει.

Η συνάρτηση recordcount επιστρέφει τον αριθμό των εγγραφών που υπάρχουν, και τέλος η resno επιστρέφει τον αριθμό εγγραφής αν τη διαβάσει ο χρήσης ή τον μετακινεί στην εγγραφή που ορίζει. π.χ.

TblCustomer.recordcount `επιστρέφει 1

TblCustomer.resno `επιστρέφει 1

x=TblCustomer.locate("code","01-001") 'x=true ή false ανάλογα με το αν βρεθεί η εγγραφή και μεταφορά στην συγκεκριμένη εγγραφή

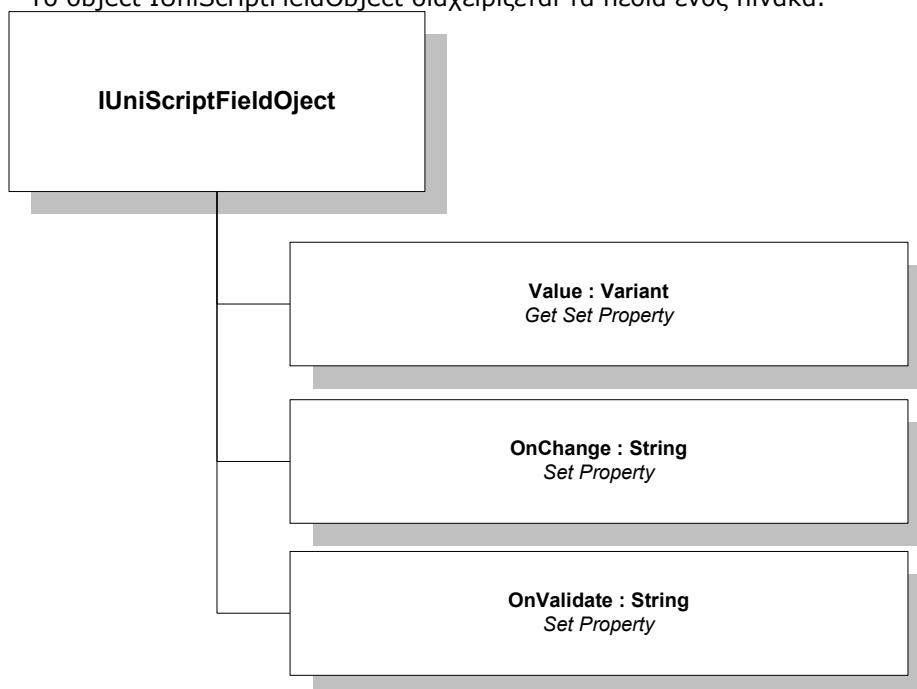
Το Property IField δίνει την τιμή ενός πεδίου μιας εγγραφής. π.χ.

Set fldCusName=tblCustomer.IField("NAME")

Η μεταβλητή fldCusName θα πάρει την τιμή του πεδίου NAME του πίνακα CUSTOMER.

4.7. IUniScriptFieldObject

Το object IUniScriptFieldObject διαχειρίζεται τα πεδία ενός πίνακα.



Οι ιδιότητες του Object IUniScriptFieldObject είναι πολύ απλές και περιλαμβάνουν την τιμή του πεδίου value και δύο event.

Η τιμή του πεδίου είναι και το βασικό property οπότε η σύνταξη

```
FldCusName = "Κάτι" είναι ισοδύναμη με την
```

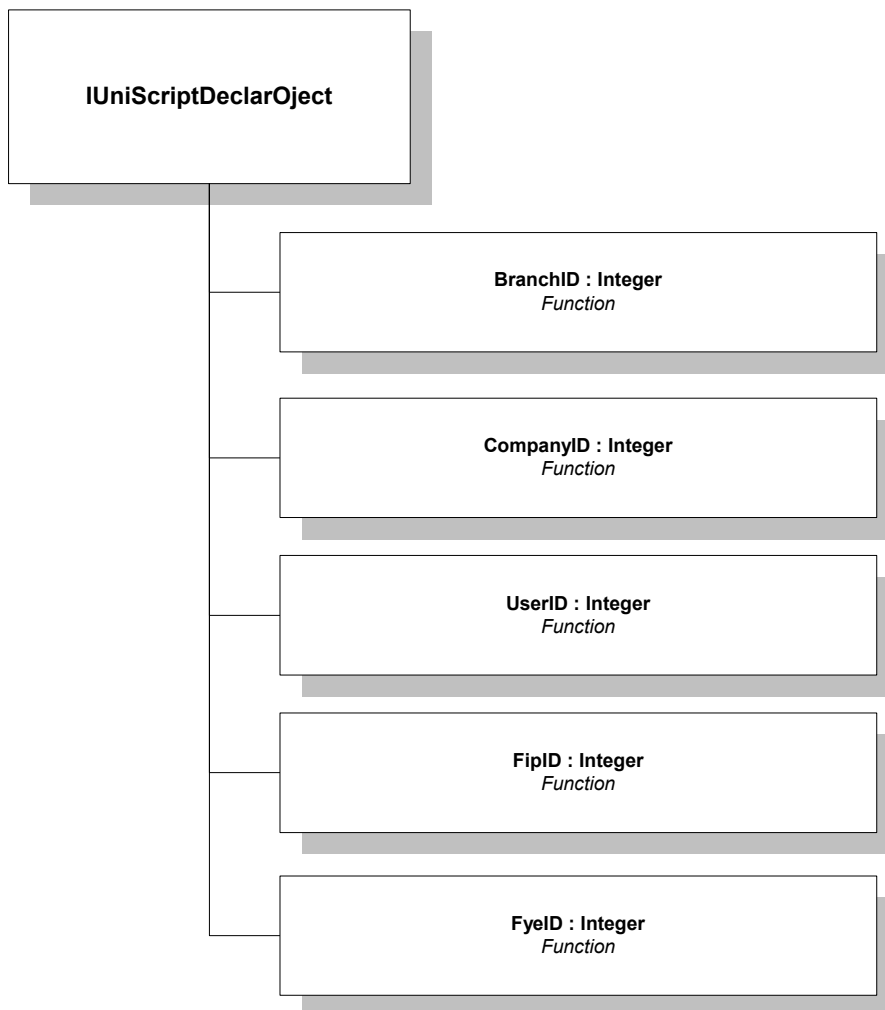
```
FldCusName.value = "Κάτι"
```

Το event OnChange ενεργοποιείται όταν χάνει το focus το πεδίο το οποίο ελέγχουμε και έχει αλλάξει η τιμή του.

Το event OnValidate ενεργοποιείται όπως και το onchange.

4.8. IUniscrptDeclarObject

Το Object IUniscrptDeclarObject είναι πληροφοριακό.



Αν χρησιμοποιήσουμε τις παρακάτω εντολές θα έχουμε:

```
Dim Myvars  
set Myvars =IclientDM.IDeclar  
msgbox myvars.comanyID
```

ένα message box το οποίο θα μας δίνει το ID της εταιρείας στην οποία έχουμε κάνει login εκείνη τη στιγμή.
Το object IclientDM.IDeclar είναι τύπου IUniscryptDeclarObject.

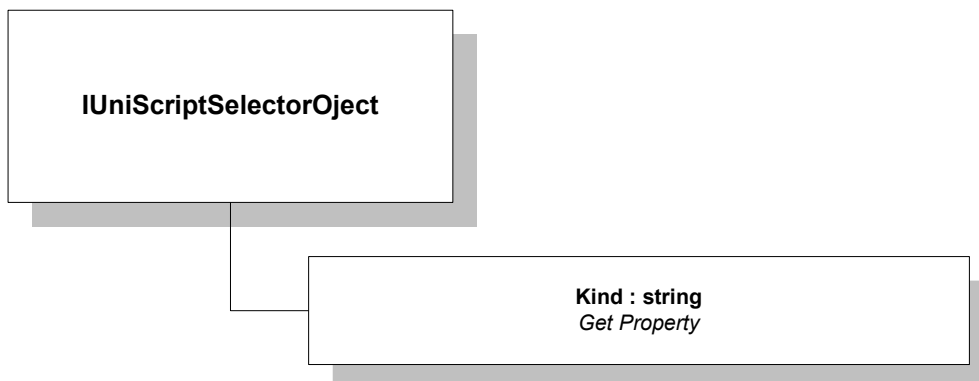
Οι κωδικοί σημαίνουν:

- **BranchID:** Το υποκατάστημα στο οποίο κάναμε login.
- **CompanyID:** Η εταιρεία στην οποία κάναμε login.
- **UserID:** Ο κωδικός χρήστη με τον οποίο κάναμε Login.
- **FipId:** Η τρέχουσα περίοδος στην οποία εργαζόμαστε.
- **FyeID:** Η τρέχουσα χρήση στην οποία εργαζόμαστε.

4.9. IUniScriptSelectorObject

Το object IUniScriptSelectorObject έχει το όνομα του selector της φόρμας.

Έχει ένα property το Kind μορφής String (κειμένου) το οποίο ο χρήστης μπορεί να διαβάσει τον Selector που χρησιμοποιείται ή να τον αλλάξει.



4.10. IUniScriptDBObject

Το IUniScriptDBObject object είναι υπεύθυνο για την αλληλεπίδραση της Script γλώσσας με τον Server.

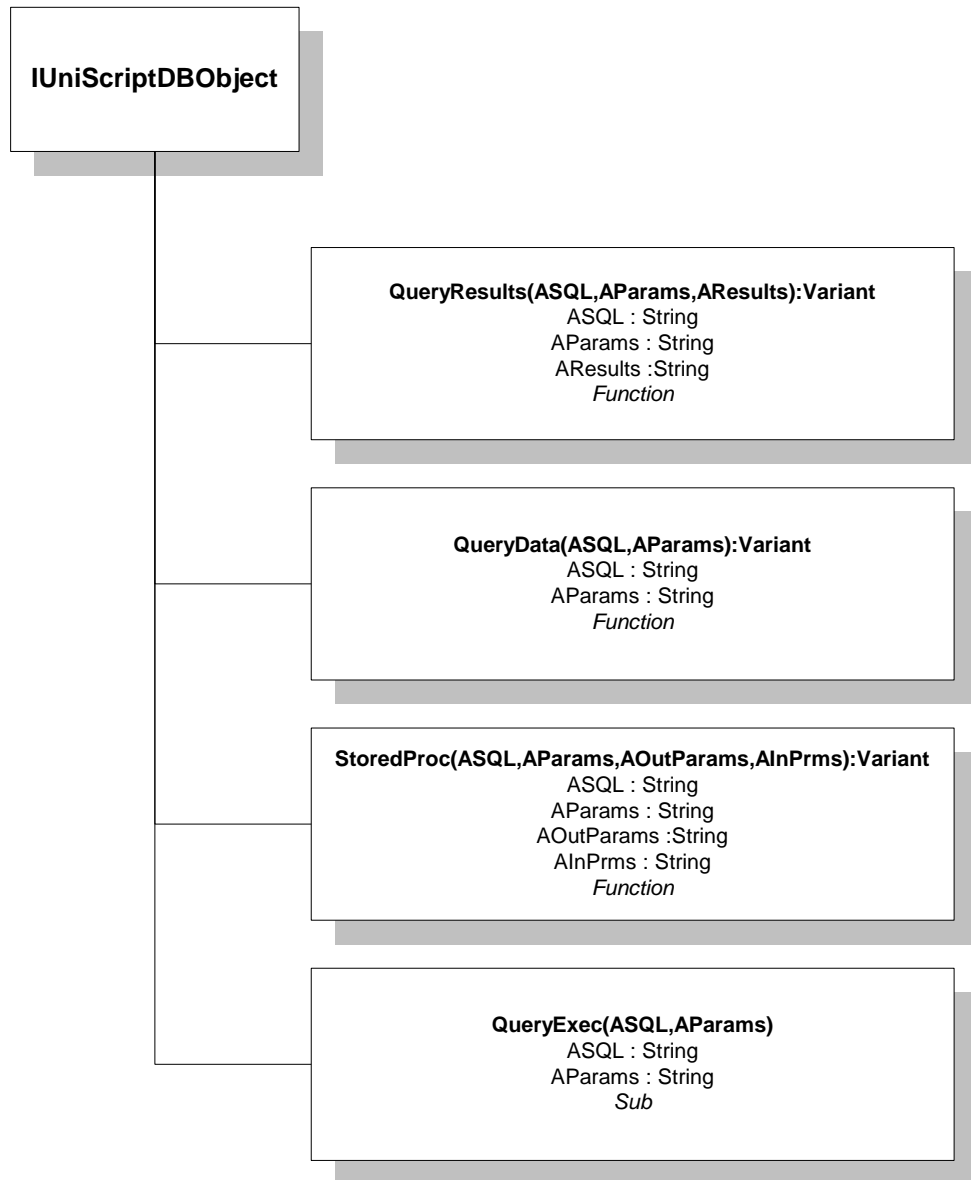
Η συνάρτηση QueryResult δίνει από μια εγγραφή έναν πίνακα με τις τιμές όπως τις έχει ζητήσει ο χρήστης από τις παραμέτρους AParams βάση ενός Select που έχει βάλει στην ASQL παράμετρο.

Αν π.χ. το AParams έχει την τιμή "Data1:Data2" θα επιστρέψει έναν πίνακα με το Data1 στη θέση 0 και το Data2 στην θέση 1 του πίνακα που θα επιστρέψει.

Η συνάρτηση QueryData επιστρέφει τα δεδομένα ενός Select που έχει βάλει ο χρήστης στην ASQL παράμετρο. Χρησιμοποιείται με τις CreateUniTable και CreateMemTable του IClientDM και IServerDM και γεμίζει με data τους πίνακες.

Η συνάρτηση QueryExec εκτελεί μια SQL εντολή ASQL με τις παραμέτρους AParams.

Τέλος η συνάρτηση StorProc εκτελεί την Stored Procedure AProcName με τις παραμέτρους AInParams και επιστρέφει ένα Recorset.



5. Παραδείγματα Χρήσης των UniScript Objects

5.1. Σενάριο 1°:

Υπάρχει ένα custom table με το όνομα TempTable και δύο πεδία τα Tempid number, TempDescr Varchar2(35). Ο χρήστης επιθυμεί να τηρεί σ' αυτόν τον πίνακα το ID και το όνομα του πελάτη, και βέβαια να ενημερώνεται με αλλαγές, διορθώσεις και διαγραφές. Ο κώδικας που κάνει τα παραπάνω είναι:

```
1 option explicit
2 Dim tblCustomer, FldCusName, FldCusCode, CusOccupation , tp1
3 dim fileobj, txt, myfile
4 dim tbltmp, fldid, flddescr
5 dim sdm, sdb
6 Sub UniSrvInitialization
7 set sdm=iserverdm
8 set sdb=sdm.IDB
9 Set tblCustomer=sDM.IsrvTable("CUSTOMER")
10 Set FldCusCode=tblCustomer.IField("ID")
11 Set CusOccupation=tblCustomer.IField("OCCUPATION")
12 Set FldCusName=tblCustomer.IField("NAME")
12 set tbltmp = sdm.createunitable("temptable")
14 tbltmp.data = sdb.querydata("Select * from temptable",null)
15 sDM.OnUpDate=("updatename")
16 sDM.Oninsert=("insertname")
17 sDM.Ondelete=("deletename")
18 set fldId =tbltmp.ifielid("tempID")
19 set fldDescr=tbltmp.ifielid("tempDescr")
20 End Sub
21 sub updatename(AHNDL)
22 dim z
23 fldid.value=fldcuscode.value
24 flddescr.value=fldcusname.value
25 z=sdb.queryresults("select tempid from temptable where
tempid =:1" _
26 , fldid.value,"tempid")
27 if not isnull(z) then
28 z=sdb.queryexec("update temptable set tempdescr = '" &
flddescr.value _
29 & "' where tempid =" & fldid.value , null)
30 end if
31 end sub
```

```

32 sub deletename (AHNDL)
33   dim z
34   fldid.value=fldcuscode.value
35   flddescr.value=fldcusname.value
36   z=sdb.queryresults("select tempid from temptable where
tempid =:1" _
37                       , fldid.value,"tempid")
38   z=sdb.queryexec("delete from temptable where tempid =" &
fldid.value 39                _ , null)
40
41 end sub
42 sub insertname (AHNDL)
43   dim z
44   fldid.value=fldcuscode.value
45   flddescr.value=fldcusname.value
46   z = sdb.queryresults("select tempid from temptable where
tempid =:1", fldid.value,"tempid")
47   if isnull(z) then
48     z = sdb.queryexec("insert into temptable values (" &
fldid.value", "' & flddescr.value & "' )", null)
49   end if
50 end sub

```

Ανάλυση του κώδικα:

Το πρόγραμμα αυτό ενεργοποιείται στον Server γιατί ο χρήστης επιθυμεί να εκτελεί κάποιες λειτουργίες όταν ενημερώνεται η Master Βάση των πελατών.

Δηλαδή οι επιπλέον εντολές που χρειάζεται θα ενεργοποιούνται κατά τη διάρκεια των εντολών "καταχώρηση" και "διαγραφή".

Η πρώτη γραμμή υποχρεώνει τη VBScript να ελέγχει αν υπάρχουν όλες οι μεταβλητές. (Μας βοηθά αν έχουμε γράψει λάθος κάποια μεταβλητή με το ανάλογο μήνυμα)

Στις γραμμές 2-5 δηλώνουμε τις μεταβλητές που θα χρησιμοποιήσουμε.

Η subroutine UniSrvInitialization 6-20 είναι η βασική ρουτίνα και εκτελείται κάθε φορά που ενεργοποιούμε την φόρμα για να δώσει τις αρχικές τιμές.

Στις εντολές 7 και 8 ορίζουμε τα βασικά object που χρησιμοποιούμε σ' όλο το πρόγραμμα.

Στην εντολή 8 ορίζουμε τον πίνακα Customer στην μεταβλητή tblCustomer.

Στις εντολές 9-12 αντιστοιχούμε πεδία του πίνακα σε ανάλογες μεταβλητές.

Στην εντολή 13 αντιστοιχούμε τον πίνακα TempTable στην μεταβλητή tbltbl χωρίς όμως να έχει δεδομένα, τα οποία τα μεταφέρουμε στην μνήμη με την εντολή 14, εκτελώντας το ανάλογο άλογο query.

Οι εντολές 15-17 αντιστοιχούν τα event του Server να εκτελούν τις ανάλογες ρουτίνες κάθε φορά που θα ενεργοποιούνται.

Τέλος οι εντολές 19 και 20 αντιστοιχούν σε μεταβλητές τα πεδία του TempTable.

Έτσι με την παραπάνω ρουτίνα έχουμε δηλώσει τον πίνακα των πελατών και τον προσωρινό πίνακα και ενεργοποιούμε τρεις υπορουτίνες κάθε φορά που γίνεται ένα από τα event του Server.

Η subroutine updatename ενεργοποιείται με το event onUpDate δηλαδή κάθε φορά που πατάμε «καταχώρηση» στην φόρμα του πελάτη.

Στις εντολές 23-24 μεταφέρει τα πεδία ID και NAME στις αντίστοιχες μεταβλητές του προσωρινού πίνακα.

Στην εντολή 25 βλέπουμε αν υπάρχει καταχωρημένος ο πελάτης στον προσωρινό πίνακα. Αν δεν είναι η μεταβλητή z θα πάρει την τιμή null. Έτσι ελέγχουμε αν η μεταβλητή z δεν είναι null για να εκτελέσουμε ένα update query στον προσωρινό πίνακα.

Η subroutine Insertname εκτελεί ακριβώς την ίδια διαδικασία και αν δεν υπάρχει ο πελάτης, εκτελεί μια Insert εντολή.

Ομοίως η DeleteName διαγράφει έναν πελάτη, χωρίς όμως να ελέγχει αν υπάρχει ο πελάτης. Δεν είναι απαραίτητος ο έλεγχος γιατί αν δεν υπάρχει ο πελάτης δεν θα υπάρχει πρόβλημα, και οποιοδήποτε μήνυμα λάθους θα περάσει στην μεταβλητή z.

5.2. Σενάριο 2^ο

Ο παραπάνω προσωρινός πίνακας πρέπει να φορτώνεται στην μνήμη και να μεταφέρει το όνομα του πελάτη στο επάγγελμα κάθε φορά που θα αλλάζει το όνομα του πελάτη.

```
1 option explicit

2 Dim tblCustomer, FldCusName, FldCusCode, CusOccupation , tp1
3 dim fileobj, txt, myfile
4 dim tbltmp, fldid, flddescr
5 dim dm, db

6 Sub UniCliInitialization
7   set dm=iclientdm
8   set db=dm.IDB
9   Set tblCustomer=IClientDM.ICliTable("CUSTOMER")
10  Set FldCusCode=tblCustomer.IField("ID")
11  Set CusOccupation=tblCustomer.IField("OCCUPATION")
12  Set FldCusName=tblCustomer.IField("NAME")
13  set tp1 =IClientDM.IDeclar
14  fldCusName.OnValidate="myCheck"
15  set tbltmp = dm.createunitable("temptable")
16  tbltmp.data = db.querydata("Select * from temptable",null)
17 End Sub
```

```

18 Sub MyCheck (AHNDL)
19   dim z
20   set fldId    =tbltmp.ifield("tempID")
21   set fldDescr=tbltmp.ifield("tempDescr")
22   flddid.value=fldcuscode.value
23   flddescr.value=fldcusname.value
24   dofind(z)
25   cusoccupation.value = flddescr.value
26 End Sub

31 sub dofind(zz)
32   dim ok
33   ok=false
34   tbltmp.first
35   do until ok
36     if zz=flddid.value then ok=true else tbltmp.next
37     if tbltmp.eof = true then ok=true
38   loop
39 end sub

```

Το πρόγραμμα αυτό ενεργοποιείται στον Client γιατί ο χρήστης επιθυμεί την ώρα που αλλάζει κάτι στο πρόγραμμα να γίνονται και οι ανάλογες αλλαγές στα αντίστοιχα πεδία.

Στην βασική ρουτίνα UniCliInitialization ορίζει τις μεταβλητές και ενεργοποιείται το πεδίο Name να ενεργοποιεί την MyCheck σε κάθε αλλαγή του.

Η ρουτίνα Mycheck ενεργοποιεί τα πεδία του προσωρινού πίνακα 20-21 και μεταφέρει εκεί τις τιμές του πελάτη 22-23. Με την εντολή 24 εκτελούμε μια σειριακή ανεύρεση στην ρουτίνα DOFIND η οποία σταματά όταν βρει τον πελάτη. Τέλος μεταφέρεται η τιμή του πεδίου DESCR του προσωρινού πίνακα στο επάγγελμα του πελάτη.

Η ρουτίνα DOFIND αρχικά πηγαίνει στην πρώτη εγγραφή του προσωρινού πίνακα και μετά προχωρά εγγραφή εγγραφή μέχρι να βρει τον πελάτη. Αν ο πίνακας είναι ταξινομημένος μπορεί να γίνει και Binary Search. (Αυτό μπορούμε να το ορίσουμε με Order By στο Select που διαβάζει τον πίνακα).

5.3. Σενάριο 3°

Θέλουμε να αλλάζουμε τα περιεχόμενα των γραμμών παραστατικού όταν αλλάξει κάτι στον Header, πχ αλλαγή τρόπου εξόφλησης.

```
1 option explicit

2 dim tblhead, fldhead , fldtype
3 dim dm,db

4 Sub UniCliInitialization
5 set dm=iclientdm
6 set db=dm.idb
7 set tblhead=dm.iclitable("fintrade")
8 set fldhead=tblhead.ifield("fttid")
9 fldhead.onchange="Mycheck"
10 end sub

11 sub mycheck(AHNDL)
12 dim ok ,i ,cnt
13 dim tblline, fldline
14 set tblline=dm.iclitable("Storetradelines")
15 set fldline=tblline.ifield("Price")
16 if isnull(fldline.value) then
17     ok=true
18 else
19     ok = false
20 end if
21 tblline.first
22 i=0
23 do until ok
24     fldline.value=fldline.value*(1.4)
25     tblline.next
26     if tblline.eof then ok =true
27     i=i+1
28 loop
29 end sub
```

Η αρχική εργασία που κάνει ο χρήστης στο παραπάνω πρόγραμμα είναι να ενεργοποιήσει τους ελέγχους, ώστε να μπορεί να κάνει τις αλλαγές στις γραμμές παραστατικού.

Στην ρουτίνα αλλαγής γραμμών myCheck, διαβάζει πρώτα τα περιεχόμενα των γραμμών και αν υπάρχει (isnull) το πεδίο τιμής, συνεχίζει (ok=true). Μετά εκτελείται ένα loop μέχρις ότου να φτάσει ο πίνακας στο τέλος του αρχείου και αλλάζει την τιμή στις αξίες των γραμμών παραστατικού.

Όταν συνδέει τον πίνακα StoreTradeLines με την μεταβλητή tblLine τα περιεχόμενα του πίνακα είναι αυτά που έχουν επιλεγεί για το συγκεκριμένο παραστατικό, και μόνο αυτά.

5.4. Σενάριο 4°

Χρησιμοποιείται ένας πίνακας (πχ TempTable) στην μνήμη και με το κλειδί TempId παίρνουμε μια τιμή την οποία χρησιμοποιούμε για κάποιο πεδίο της φόρμας εισαγωγής

```
1 option explicit
2 Dim tblCustomer,FldCusCode,webpage ,tp1
3 dim fileobj, txt, myfile
4 dim tbltmp, fldid, flddescr
5 dim dm,db

6 Sub UniCliInitialization
7 set dm=iclientdm
8 set db=dm.IDB
9 Set tblCustomer=IClientDM.ICliTable("supplier")
10 Set FldCusCode=tblCustomer.IField("email")
11 Set webpage=tblCustomer.IField("webpage")
12 set tp1 =IClientDM.IDeclar
13 fldCusCode.OnValidate="myCheck"
14 set tbltmp = dm.createunitable("temptable")
15 tbltmp.data = db.querydata("Select * from temptable",null)
16 set fldId =tbltmp.ifield("tempID")
17 set fldDescr=tbltmp.ifield("tempDescr")
18 End Sub

20 Sub MyCheck(AHNDL)
21 if isnumeric(fldcuscode.value) then
22 webpage.value = dofind(fldcuscode.value)
23 else
24 webpage.value = "Error number"
25 end if
26 End Sub
```

```

27 function dofind(zz)
28 dim ok , found ,i
28 i=1
29 ok=false
30 found=false
31 tbltmp.first
32 do
33   if cint(zz)=cint(fldid.value) then
34     ok=true
35   else
36     tbltmp.next
37   end if
38 loop until (ok = true) or (tbltmp.eof)
39 if ok then
40   dofind=flddescr.value
41 else
42   dofind="Wrong code"
43 end if
44 end function

```

Το παραπάνω πρόγραμμα λειτουργεί στο Client των προμηθευτών.

Φορτώνει τον πίνακα TempTable στην μνήμη κάθε φορά που ο χρήστης ανοίγει τη φόρμα και ενεργοποιείται με το πεδίο e-mail.

Αν δώσει έναν υπαρκτό κωδικό στο e-mail το script ψάχνει και βρίσκει την αντίστοιχη περιγραφή του πίνακα και επιστρέφει την ανάλογη τιμή στο Web address.

5.5. Σενάριο 5°

Δημιουργία μιας Master Form η οποία θα έχει έναν κύριο πίνακα, τον πίνακα δημιουργίας αριθμών αριθμημένων επιταγών, και έναν πίνακα Detail ο οποίος θα έχει τους αριθμούς για περαιτέρω επεξεργασία.

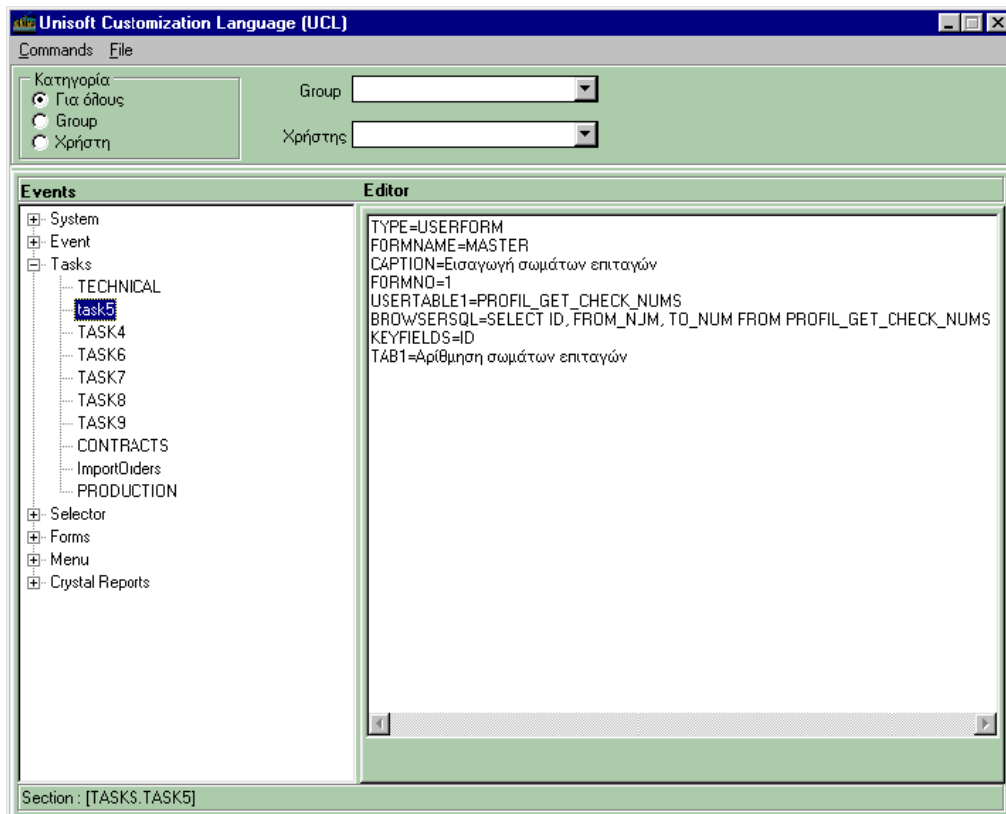
Header πίνακας

```
CREATE TABLE PROFIL_GET_CHECK_NUMS (  
  ID          NUMBER (10)  NOT NULL,  
  FROM_NUM    VARCHAR2 (10) NOT NULL,  
  TO_NUM      VARCHAR2 (10) NOT NULL,  
  ACCID       NUMBER (10),  
  CREATED     CHAR (1),  
  CHECK_DIG   NUMBER (1),  
  X_VALUE     NUMBER (1))
```

Details πίνακας

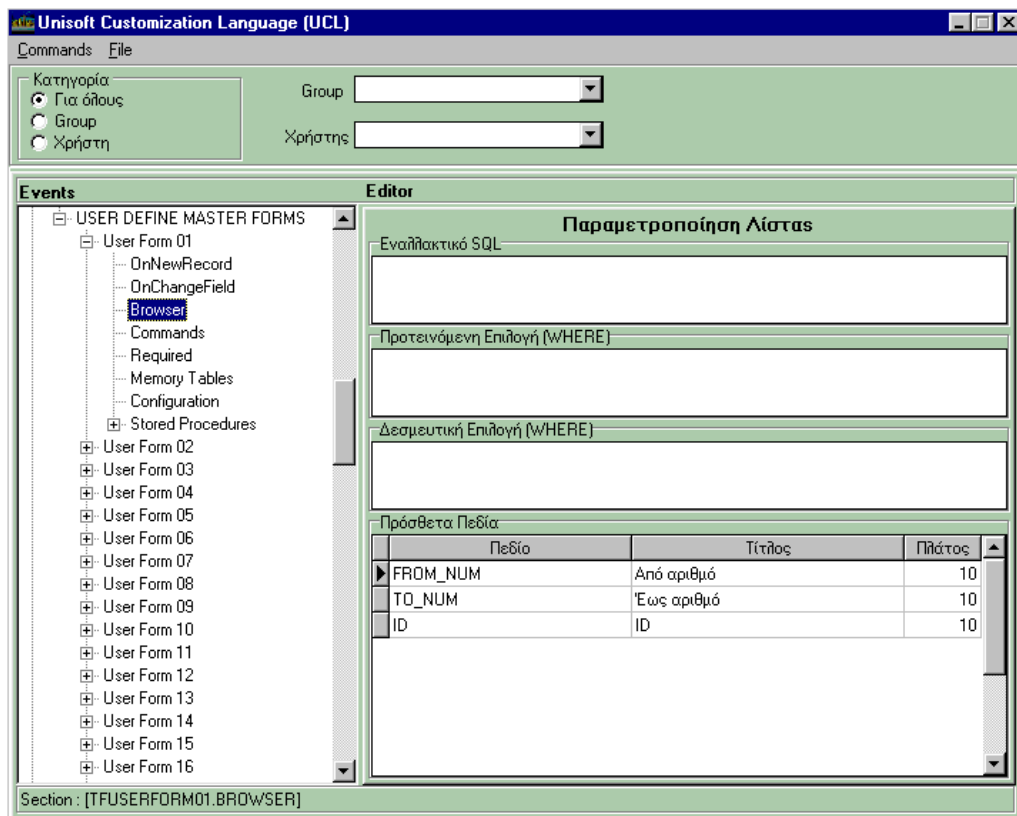
```
CREATE TABLE PROFIL_CHECK_NUMBERS (  
  ID          NUMBER (10)  NOT NULL,  
  BANKACCOUNT VARCHAR2 (50) NOT NULL,  
  CHECK_NUMER VARCHAR2 (50) NOT NULL,  
  BODY        NUMBER (10),  
  CHECK_ST    NUMBER (1),  
  CREID       NUMBER (10))
```

Δημιουργούμε στο περιβάλλον του UCL ένα task Master Form με τα παρακάτω χαρακτηριστικά



Με την επιλογή FORMNO=1 ορίζουμε ότι θα χρησιμοποιηθεί η User Defined Form No 1.

Ορίζουμε το Browser ως εξής:



Και στο Configuration Section υπάρχουν οι δηλώσεις:

```
USERTABLE2=PROFIL_CHECK_NUMBERS;CREID;ID;PROFIL_GET_CHECK_NUMS  
;ID;1;1;1
```

Όπου το Details table συνδέεται με το βασικό με το πεδίο CREID με σχέση πολλά προς ένα στο πεδίο ID του κύριου πίνακα.

Τέλος, συνδέουμε το σχετικό Task σε μια επιλογή του MENU.

Καταχωρούνται οι αλλαγές στο UCL και περνάμε στην επόμενη διαδικασία που είναι ο σχεδιασμός της φόρμας.

The screenshot shows a window titled "Εισαγωγή σωμάτων επιταγών [Customize]". It features a form with two input fields: "Από αριθμό" (From number) and "Έως αριθμό" (To number). The "Έως αριθμό" field contains the text "UDesignCtrl7". There are also two small buttons labeled "UD" and "X UD". Below the form is a table with the following columns: "Λογαριασμός", "Αρ. Επιταγής", "Σώμα", and "ID". The table is currently empty.

Με τον σχεδιασμό της ολοκληρώνεται η διαδικασία ελέγχου της φόρμας από το Atlantis και μένει η λογική που χρειάζεται από τη VBSCRIPT.

Στις Custom Forms το Atlantis φορτώνει πρώτα τη φόρμα, κατόπιν το VBSCRIPT και τέλος τη σχετική Class.

Έτσι ο μόνος έλεγχος σε τέτοιες φόρμες που μπορούμε να έχουμε είναι των event του Server, onInsert, onDelete και onUpdate.

Ο κώδικας έχει υλοποιηθεί στο τμήμα Server...

OPTION EXPLICIT

```
dim idm,idb
dim tblget,tblcheck,tblacc
dim fldfrom, fldto, fldacc, fldcr, fldid, fldcd
dim fldx, check_dig, x_value

Sub UniSrvInitialization
    set idm=iserverdm
    set idb=idm.idb
    idm.onupdate="insertsub"
    idm.oninsert="insertsub"
    set tblget=idm.isrvtable("PROFIL_GET_CHECK_NUMS")
    set fldid =tblget.iframe("ID")
End Sub

sub insertsub(AHNDL)
    dim x,xlen,ok,insstr,insvar,getbd ,cr
    cr = chr(10)
    set fldfrom=tblget.iframe("FROM_NUM")
    set fldto =tblget.iframe("TO_NUM")
    set fldacc =tblget.iframe("ACCID")
    set fldcr =tblget.iframe("CREATED")
    set fldcd =tblget.iframe("check_dig")
    set fldx =tblget.iframe("x_value")
    check_dig=-1 : x_value=fldx.value
    ok = isnumeric(fldfrom.value) and isnumeric(fldto.value)
    if fldcr.value = "Y" then ok = false
    if ok then
        dim fld2id,fld2ba,fld2cn,fld2bd,fld2cs,fld2ci ,xend,z
        insstr = "insert into PROFIL_CHECK_NUMBERS values ("
        fld2ba = idb.queryresults("select BACCNUMBER FROM
        BANKACCOUNT WHERE ID=" _
                                &
        fldacc.value,null,"BACCNUMBER")
        fld2ba = "" & fld2ba & ""
        fld2cs = "null"
        fld2bd = "0"
        fld2ci = fldid.value
        xlen = len(fldfrom.value)
        x = cdbl(fldfrom.value)
```

```

xend = cdbl(fldto.value)
do until x > xend
    fld2cn=calcbody(x,xlen)
    fld2id = idb.queryresults("select PROFIL_CHECK_NUMBERS_SEQ.nextval
from dual" ,null,"nextval")
    insvar=fld2id & "," & fld2ba & "," & fld2cn & "," & fld2bd &
"," & fld2cs & "," & fld2ci
    z=idb.queryexec(insstr & cr & insvar & "), null)
    x=x+1
loop
fldcr.value="Y"
z=idb.queryexec("update PROFIL_GET_CHECK_NUMS set created ='Y'
where id = " & fldid.value , null )
else
doerror "Ìà äðéôáððÐ äéääééáóá"
end if
end sub

```

```

function calcbody(x,xl)
dim i,i1,i2
i1 = len(cstr(x))
i2 = cint(xl)
i = i2-i1
if i > 0 then
calcbody = string(i,48)
end if
calcbody = "" & calcbody & cstr(x)
if check_dig=-1 then
check_dig=fldcd.value
else
if check_dig>0 then
check_dig=check_dig+1
if check_dig = 10 then check_dig=0
else
if x_value=1 then
check_dig=check_dig+1
if check_dig=10 then check_dig=0
x_value=0
else
x_value=1

```

```

        end if
    end if
end if
    calcbody=calcbody & "-" & check_dig & ""
end function

```

Ο παραπάνω κώδικας, όταν εισάγεται αρχική και τελική επιταγή υπολογίζει αυτόματα τους αριθμούς επιταγών και δημιουργεί τις ανάλογες εγγραφές.

5.6. Σενάριο 6°

Ένα αρχείο θα δέχεται μια συγκεκριμένη σειρά από τις παραγγελίες πωλήσεων και δημιουργεί εγγραφές παραγωγής για τα στάδια που ακολουθούνται σε μια εταιρία.

Master Table

```

CREATE TABLE PROFIL_ORDERS (
    FRTID      NUMBER (10)  NOT NULL,
    CUSID      NUMBER (10)  NOT NULL,
    CUSNAME    VARCHAR2 (35),
    TRADECODE  VARCHAR2 (15),
    ID         NUMBER (10),
    DUMMY      NUMBER (10))

```

Details Table

```

CREATE TABLE PROFIL_ORDER_LINES (
    ID         NUMBER (10)  NOT NULL,
    FTRID     NUMBER (10)  NOT NULL,
    LINES     NUMBER (10),
    QTY1      FLOAT,
    QTY2      FLOAT,
    QTY3      FLOAT,
    QTY4      FLOAT,
    QTY5      FLOAT,
    QTY6      FLOAT,
    QTY7      FLOAT,
    QTY8      FLOAT,
    QTY9      FLOAT,
    QTY10     FLOAT,
    QTY11     FLOAT,
    QTY12     FLOAT,
    QTY13     FLOAT,
    QTY14     FLOAT,
    QTY15     FLOAT,

```

```

QTY16    FLOAT,
QTY17    FLOAT,
QTY18    FLOAT,
QTY19    FLOAT,
QTY20    FLOAT,
ITEM     VARCHAR2 (30),
QTYMAX   FLOAT,
ADDVAL   FLOAT)

```

Δημιουργείται η σχετική φόρμα (και για μεγαλύτερο μέγεθος δηλώνονται διαστάσεις στο Configuration Section της φόρμας:

```

USERTABLE4=PROFIL_ORDER_LINES;FTRID;ID;PROFIL_ORDERS;ID;1;1;1
SELF.LEFT=5
SELF.WIDTH=900

```

όπου το left και το width είναι οι διαστάσεις του παραθύρου.

Η φόρμα έχει τη μορφή:

Ο κώδικας ενημερώνει μόνο κάθε εγγραφή και δεν δημιουργεί νέες.

```

option explicit
dim dm,db
Sub UniSrvInitialization
    set dm=iserverdm
    set db=dm.idb
    dm.oninsert="oninsert"
    dm.onupdate="onupdate"
End Sub

```

```

sub oninsert(AHNLD)
  doerror "Δεν επιτρέπεται η καταχώρηση νέας εγγραφής"
end sub
sub onupdate(AHNLD)
  dim ordtbl, ordid, orddum
  dim orltbl,
  orlq1, orlq2, orlq3, orlq4, orlq5, orlq6, orlq7, orlq8, orlq9, orlq10
  dim
  orlq11, orlq12, orlq13, orlq14, orlq15, orlq16, orlq17, orlq18, orlq19
, orlq20
  dim orladd, orlmax, orlid, orlitem, amm, z
  set ordtbl=dm.isrvtable("profil_orders")
  set orltbl=dm.isrvtable("Profil_order_lines")
  set ordid=ordtbl.ifield("id")
  set orddum=ordtbl.ifield("dummy")
  set orlq1=orltbl.ifield("qty1")
  set orlq2=orltbl.ifield("qty2")
  set orlq3=orltbl.ifield("qty3")
  set orlq4=orltbl.ifield("qty4")
  set orlq5=orltbl.ifield("qty5")
  set orlq6=orltbl.ifield("qty6")
  set orlq7=orltbl.ifield("qty7")
  set orlq8=orltbl.ifield("qty8")
  set orlq9=orltbl.ifield("qty9")
  set orlq10=orltbl.ifield("qty10")
  set orlq11=orltbl.ifield("qty11")
  set orlq12=orltbl.ifield("qty12")
  set orlq13=orltbl.ifield("qty13")
  set orlq14=orltbl.ifield("qty14")
  set orlq15=orltbl.ifield("qty15")
  set orlq16=orltbl.ifield("qty16")
  set orlq17=orltbl.ifield("qty17")
  set orlq18=orltbl.ifield("qty18")
  set orlq19=orltbl.ifield("qty19")
  set orlq20=orltbl.ifield("qty20")
  set orladd=orltbl.ifield("addval")
  set orlmax=orltbl.ifield("qtxmax")
  set orlid =orltbl.ifield("id")
  set orlitem=orltbl.ifield("item")
  if isnull(orddum.value) then
    msgbox "Δεν έχει επιλεγεί παραγγελία"
  else
    orltbl.first
    do
      select case orddum.value
        case 1
          if isnull(orldtbl.ifield("qty1").value) then orlq1.value=0
          if isnull(orldtbl.ifield("addval").value) then orladd.value=0
          if orlq1.value + orladd.value > orlmax.value then
            msgbox "Η τιμή είναι μεγαλύτερη από την
παραγγελία"
          else
            amm =cdbl(orldtbl.ifield("qty1").value) + cdbl(orldtbl.ifield("addval").value)
            z=db.queryexec("update profil_order_lines set
qty1=" & amm & _
", addval=0 where id=" &
orldtbl.ifield("id").value,null)

```

```

        end if
    case 2
        if isnull(ork2.value) then ork2.value=0
        if isnull(orkadd.value) then orkadd.value=0
        if ork2.value + orkadd.value > orkmax.value then
            msgbox " Η τιμή είναι μεγαλύτερη από την
παραγγελία "
        else
            amm =ork2.value + orkadd.value
            z=db.queryexec("update profil_order_lines set
qty2=" & amm & _
                                ", addval=0 where id=" &
orkid.value,null)
        end if
    case 3
        if isnull(ork3.value) then ork3.value=0
        if isnull(orkadd.value) then orkadd.value=0
        if ork3.value + orkadd.value > orkmax.value then
            msgbox " Η τιμή είναι μεγαλύτερη από την
παραγγελία "
        else
            amm =ork3.value + orkadd.value
            z=db.queryexec("update profil_order_lines set
qty3=" & amm & _
                                ", addval=0 where id=" &
orkid.value,null)
        end if
    case 4
        if isnull(ork4.value) then ork4.value=0
        if isnull(orkadd.value) then orkadd.value=0
        if ork4.value + orkadd.value > orkmax.value then
            msgbox " Η τιμή είναι μεγαλύτερη από την
παραγγελία "
        else
            amm =ork4.value + orkadd.value
            z=db.queryexec("update profil_order_lines set
qty4=" & amm & _
                                ", addval=0 where id=" &
orkid.value,null)
        end if
    case 5
        if isnull(ork5.value) then ork5.value=0
        if isnull(orkadd.value) then orkadd.value=0
        if ork5.value + orkadd.value > orkmax.value then
            msgbox " Η τιμή είναι μεγαλύτερη από την
παραγγελία "
        else
            amm =ork5.value + orkadd.value
            z=db.queryexec("update profil_order_lines set
qty5=" & amm & _
                                ", addval=0 where id=" &
orkid.value,null)
        end if
    case 6
        if isnull(ork6.value) then ork6.value=0
        if isnull(orkadd.value) then orkadd.value=0
        if ork6.value + orkadd.value > orkmax.value then

```

```

        msgbox " Η τιμή είναι μεγαλύτερη από την
paraγγελία "
    else
        amm =orlq6.value + orladd.value
        z=db.queryexec("update profil_order_lines set
qty6=" & amm & _
                                ", addval=0 where id=" &
orlid.value,null)
    end if
    case 7
    if isnull(orlq7.value) then orlq7.value=0
    if isnull(orladd.value) then orladd.value=0
    if orlq7.value + orladd.value > orlmax.value then
        msgbox " Η τιμή είναι μεγαλύτερη από την
paraγγελία "
    else
        amm =orlq7.value + orladd.value
        z=db.queryexec("update profil_order_lines set
qty7=" & amm & _
                                ", addval=0 where id=" &
orlid.value,null)
    end if
    case 8
    if isnull(orlq8.value) then orlq8.value=0
    if isnull(orladd.value) then orladd.value=0
    if orlq8.value + orladd.value > orlmax.value then
        msgbox " Η τιμή είναι μεγαλύτερη από την
paraγγελία "
    else
        amm =orlq8.value + orladd.value
        z=db.queryexec("update profil_order_lines set
qty8=" & amm & _
                                ", addval=0 where id=" &
orlid.value,null)
    end if
    case 9
    if isnull(orlq9.value) then orlq9.value=0
    if isnull(orladd.value) then orladd.value=0
    if orlq9.value + orladd.value > orlmax.value then
        msgbox " Η τιμή είναι μεγαλύτερη από την
paraγγελία "
    else
        amm =orlq9.value + orladd.value
        z=db.queryexec("update profil_order_lines set
qty9=" & amm & _
                                ", addval=0 where id=" &
orlid.value,null)
    end if
    case 10
    if isnull(orlq10.value) then orlq10.value=0
    if isnull(orladd.value) then orladd.value=0
    if orlq10.value + orladd.value > orlmax.value then
        msgbox " Η τιμή είναι μεγαλύτερη από την
paraγγελία "
    else
        amm =orlq10.value + orladd.value

```

```

        z=db.queryexec("update profil_order_lines set
qty10=" & amm & _
                                ", addval=0 where id=" &
orlid.value,null)
    end if
    case 11
        if isnull(orkq11.value) then orkq11.value=0
        if isnull(orkladd.value) then orkladd.value=0
        if orkq11.value + orkladd.value > orlmax.value then
            msgbox " Η τιμή είναι μεγαλύτερη από την
παραγγελία "
        else
            amm =orkq11.value + orkladd.value
            z=db.queryexec("update profil_order_lines set
qty11=" & amm & _
                                ", addval=0 where id=" &
orlid.value,null)
        end if
        case 12
            if isnull(orkq12.value) then orkq12.value=0
            if isnull(orkladd.value) then orkladd.value=0
            if orkq12.value + orkladd.value > orlmax.value then
                msgbox " Η τιμή είναι μεγαλύτερη από την
παραγγελία "
            else
                amm =orkq12.value + orkladd.value
                z=db.queryexec("update profil_order_lines set
qty12=" & amm & _
                                    ", addval=0 where id=" &
orlid.value,null)
            end if
            case 13
                if isnull(orkq13.value) then orkq13.value=0
                if isnull(orkladd.value) then orkladd.value=0
                if orkq13.value + orkladd.value > orlmax.value then
                    msgbox " Η τιμή είναι μεγαλύτερη από την
παραγγελία "
                else
                    amm =orkq13.value + orkladd.value
                    z=db.queryexec("update profil_order_lines set
qty13=" & amm & _
                                        ", addval=0 where id=" &
orlid.value,null)
                end if
                case 14
                    if isnull(orkq14.value) then orkq14.value=0
                    if isnull(orkladd.value) then orkladd.value=0
                    if orkq14.value + orkladd.value > orlmax.value then
                        msgbox " Η τιμή είναι μεγαλύτερη από την
παραγγελία "
                    else
                        amm =orkq14.value + orkladd.value
                        z=db.queryexec("update profil_order_lines set
qty14=" & amm & _
                                    ", addval=0 where id=" &
orlid.value,null)
                    end if

```



```

case 15
  if isnull(orldq15.value) then orldq15.value=0
  if isnull(orldadd.value) then orldadd.value=0
  if orldq15.value + orldadd.value > orldmax.value then
    msgbox " Η τιμή είναι μεγαλύτερη από την
    παραγγελία "
  else
    amm =orldq15.value + orldadd.value
    z=db.queryexec("update profil_order_lines set
    qty15=" & amm & _
    ", addval=0 where id=" &
    orldid.value,null)
  end if
  case 16
  if isnull(orldq16.value) then orldq16.value=0
  if isnull(orldadd.value) then orldadd.value=0
  if orldq16.value + orldadd.value > orldmax.value then
    msgbox " Η τιμή είναι μεγαλύτερη από την
    παραγγελία "
  else
    amm =orldq16.value + orldadd.value
    z=db.queryexec("update profil_order_lines set
    qty16=" & amm & _
    ", addval=0 where id=" &
    orldid.value,null)
  end if
  case 17
  if isnull(orldq17.value) then orldq17.value=0
  if isnull(orldadd.value) then orldadd.value=0
  if orldq17.value + orldadd.value > orldmax.value then
    msgbox " Η τιμή είναι μεγαλύτερη από την
    παραγγελία "
  else
    amm =orldq17.value + orldadd.value
    z=db.queryexec("update profil_order_lines set
    qty17=" & amm & _
    ", addval=0 where id=" &
    orldid.value,null)
  end if
end select
  orltbl.next
loop until orltbl.eof
end if
end sub

```

5.7. Σενάριο 7^ο (CREATE BUTTON)

Δημιουργεί ένα νέο κουμπί κάτω από το ακύρωση και εκτελεί την ίδια εργασία με το πλήκτρο ακύρωση.

```
Sub UniFrmInitialization
    set cbt = IForm.IButton("CancelBtn")
    l = cbt.propord( "Left" )
    t = cbt.PropOrd( "Top" )
    w = cbt.PropOrd( "Width" )
    h = cbt.PropOrd( "Height" )
    set ubt = IUApp.CreateButton( "DataPanel", l, t+h+3, w, h,
    "Undo" )
    ubt.onClick = "ubtclick"
End Sub

sub ubtClick
    set xbt = IForm.IButton("CancelBtn")
    xbt.Click
end sub
```

5.8. Σενάριο 8^ο (CREATE FORM)

Δημιουργεί ένα κουμπί και στην συνέχεια εμφανίζει την φόρμα των αξιόγραφων.

```
Sub UniFrmInitialization
set cbt = IForm.IButton("CancelBtn")
l = cbt.propord( "Left" )
t = cbt.PropOrd( "Top" )
w = cbt.PropOrd( "Width" )
h = cbt.PropOrd( "Height" )

set ubt = IUApp.CreateButton( "DataPanel", l, t+h+3, w, h,
"Αξιόγραφα" )
ubt.onClick = "xxClick"
End Sub

sub xxClick
    set f = IUApp.CreateForm( "TBillFRM", 0 )
    f.show( true )
end sub

CREATE DFMMFORM
Δημιουργεί ένα κουμπί και στην συνέχεια εμφανίζει Custom Φόρμα
(Δημιουργημένη με SDK)
Το DFM πρέπει να αποθηκευτεί σε κοινό φάκελο έτσι ώστε να το
βλέπουν όλοι πχ "F:\ATLADFM\unit2.dfm"
Sub UniFrmInitialization

set cbt = IForm.IButton("CancelBtn")
l = cbt.propord( "Left" )
t = cbt.PropOrd( "Top" )
w = cbt.PropOrd( "Width" )
h = cbt.PropOrd( "Height" )
```

```

set ubt = IUApp.CreateButton( "DataPanel", 1, t+h+3, w, h,
"Undo" )
ubt.onClick = "ubtclick"
End Sub

sub ubtClick
SET pelid=IForm.IClientDm.ICliTable("CUSTOMER").IFIELD("NAME")
set xfm = IUApp.CreateDFMForm( "TCBaseForm", "C:\unit2.dfm" )
xfm.PropString("Caption")= xfm.PropString("Caption") &
"Πελάτες [" & PELID.VALUE & "]"
xfm.Show( false )
end sub

ON LOCATE

Sub UniCliInitialization
    IClientDm.OnLocate= "LocCbt"
End Sub

sub LocCbt
    msgbox "test"
End Sub

```

6. Debugger προγραμμάτων εφαρμογής

6.1. Γενικά

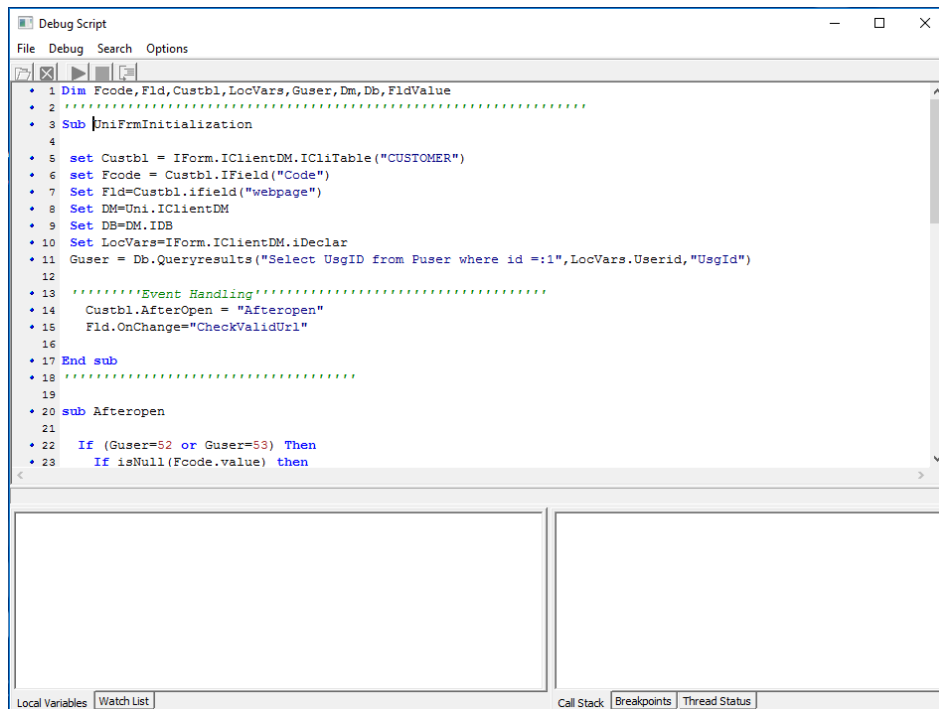
Η εφαρμογή Atlantis παρέχει στο προγραμματιστή τη δυνατότητα χρήσης ενσωματωμένου εργαλείου εντοπισμού σφαλμάτων (debugger) των προγραμμάτων που έχουν ενσωματωθεί στις φόρμες (UniFrmInitialization) και στο client datamodule (UniCliInitialization) της εφαρμογής. Το συγκεκριμένο εργαλείο μπορεί να χρησιμοποιηθεί για όλες τις υποστηριζόμενες γλώσσες προγραμματισμού (βλέπε παράγραφο 1.5).

6.2. Ενεργοποίηση Debugger

Η ενεργοποίηση του Debugger γίνεται από το προγραμματιστή πριν την έναρξη λειτουργίας της εφαρμογής Atlantis με τη προσθήκη της παραμέτρου /DebugScript στο command line. Στη συντόμευση επιφάνειας (Shortcut) του Windows από την οποία εκτελείται η εφαρμογή προσθέτουμε τη παραπάνω παράμετρο στο πεδίο «Προορισμός» (Target). Π.χ.
 "C:\Unisoft Software\Atlantis E.R.P\Go3.exe" /DebugScript

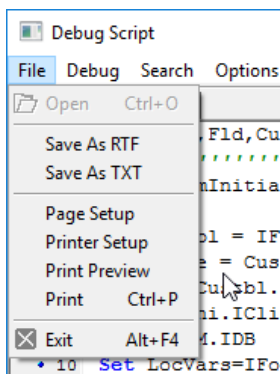
6.3. Λειτουργία Debugger

Με δεδομένο ότι η εφαρμογή Atlantis εκτελείται με τη παράμετρο /DebugScript η λειτουργία της θα σταματήσει αμέσως μόλις ο χρήστης της εφαρμογής εισέλθει σε οποιαδήποτε ενότητα υπάρχει ενεργοποιημένο πρόγραμμα (π.χ. VB Script) στη φόρμα ή στο client datamodule, και θα ανοίξει το "Debug Script" παράθυρο.

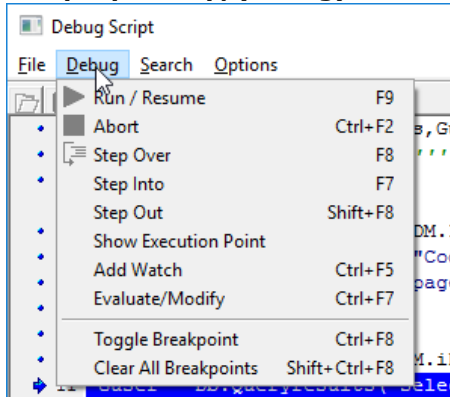


Από το σημείο αυτό και στη συνέχεια ο προγραμματιστής μπορεί να εκτελέσει όλες τις γνωστές διαδικασίες debugging ώστε να ελέγξει τη λειτουργικότητα του προγράμματός του.
Το εργαλείο υποστηρίζει λειτουργίες όπως:

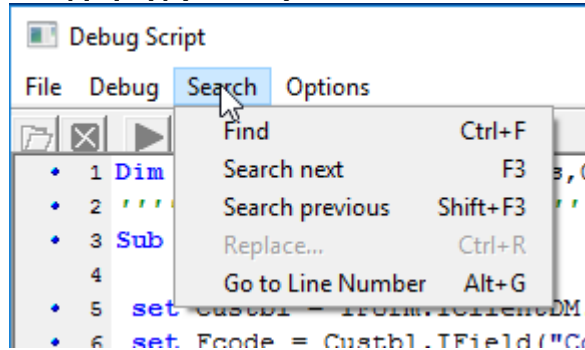
Αρχειου (File)



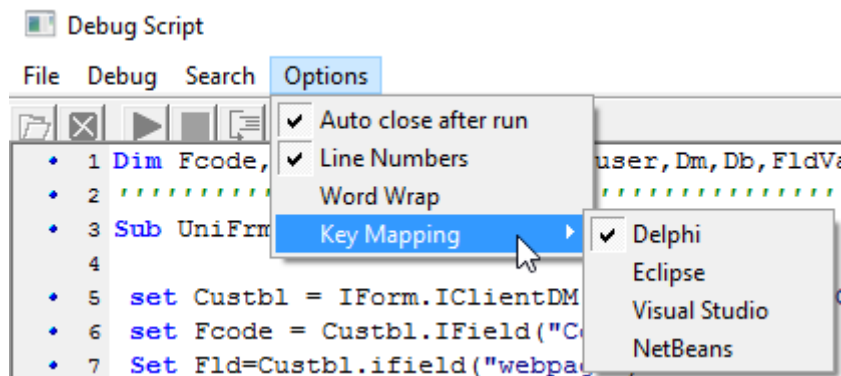
Εκπαλμάτωσης (Debug)



Αναζήτησης (Search)



Επιλογών Λειτουργίας (Options)



Τέλος υποστηρίζονται λειτουργίες όπως Watch List, Call Stack, Thread Status οι οποίες ενεργοποιούνται από τις αντίστοιχες ενότητες στο κάτω μέρος της οθόνης του debugger.

